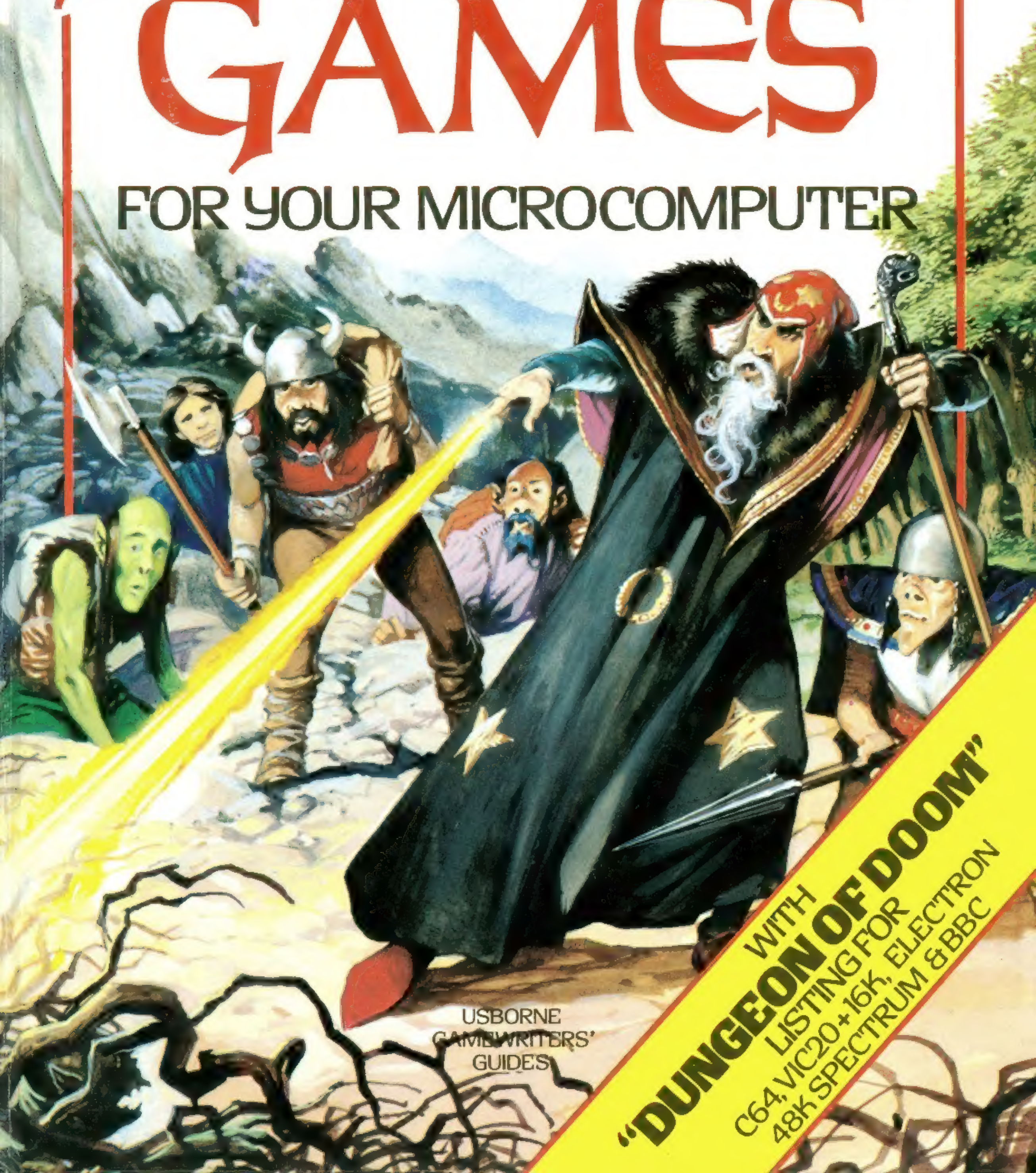
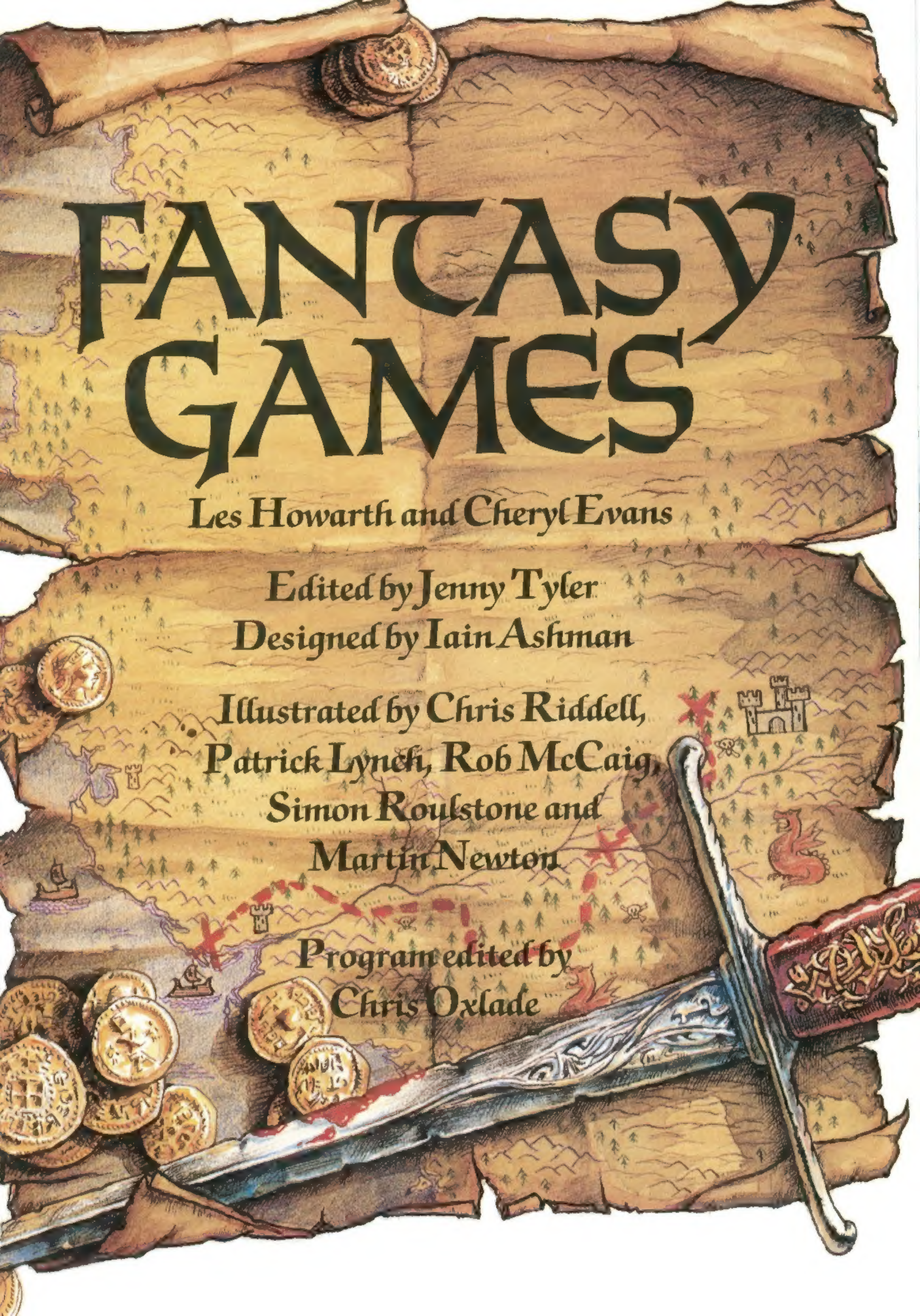
WRITE YOUR OWN FANTASY GAMES

FOR YOUR MICROCOMPUTER



USBORNE
GAMEWRITERS'
GUIDES

WITH
"DUNGEON OF DOOM"
LISTING FOR
C64, VIC20+16K, ELECTRON
48K SPECTRUM & BBC



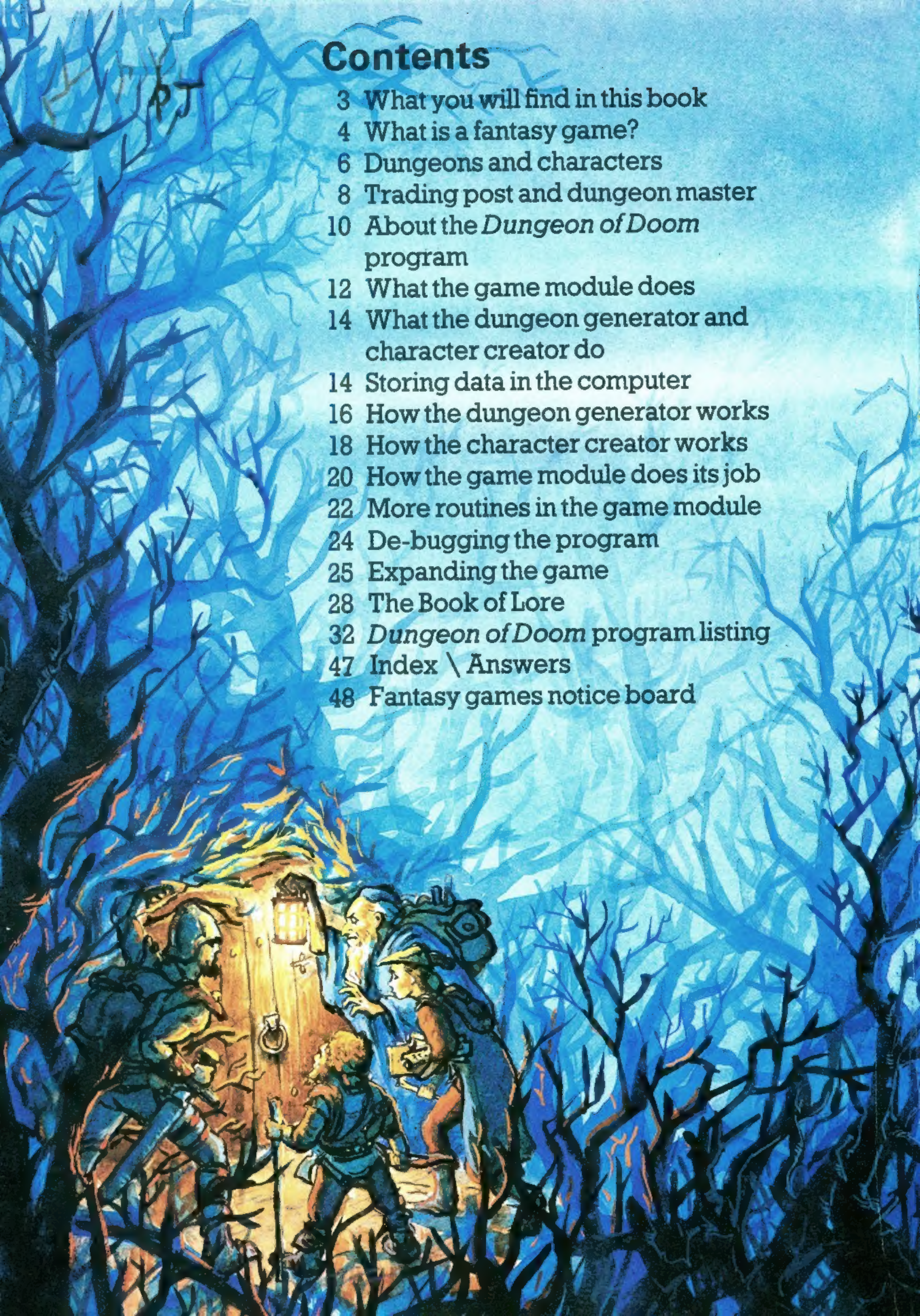
FANTASY GAMES

Les Howarth and Cheryl Evans

Edited by Jenny Tyler
Designed by Iain Ashman

Illustrated by Chris Riddell,
Patrick Lynch, Rob McCaig,
Simon Roulstone and
Martin Newton

Program edited by
Chris Oxlade



Contents

3 What you will find in this book
4 What is a fantasy game?
6 Dungeons and characters
8 Trading post and dungeon master
10 About the *Dungeon of Doom* program
12 What the game module does
14 What the dungeon generator and character creator do
14 Storing data in the computer
16 How the dungeon generator works
18 How the character creator works
20 How the game module does its job
22 More routines in the game module
24 De-bugging the program
25 Expanding the game
28 The Book of Lore
32 *Dungeon of Doom* program listing
47 Index \ Answers
48 Fantasy games notice board

What you will find in this book

This book aims to help you write fantasy games for your computer by giving you an example you can play and taking you inside the program to see how it works. If you don't know what a fantasy game is, you can find out on pages 4-9.

The *Dungeon of Doom* game

Dungeon of Doom is the name of the fantasy game written specially for this book. The rules for playing it are on pages 28-31. These pages are called the Book of Lore, as this is the name fantasy gamers usually give their notes concerning the rules and conditions in their fantasy world. There are detailed explanations of how the program works on pages 10-23. The program listing, which you can type into your own computer, is on pages 32-45. It will work on the Commodore 64, Vic, Spectrum 48K, BBC and Electron. Conversion lines are given where necessary on the listing pages.

Typing in the program

The program for *Dungeon of Doom* is very long and not easy to type in accurately, so you should be prepared to take it slowly and check it as you go along. You may find the program explanations difficult to understand at first, too, but don't be discouraged. Take your time and read them as often as you like to make sure you really follow them. It may help if you type the game into the computer first, then play it, referring to the explanations as you go along to see how the program achieves what you see on your screen.

You may prefer to play the game before finding out how it works so that you will not know what to expect. To do this, go straight to page 32 and type in the program without looking at pages 10-25. You will also need to read the story of *Dungeon of Doom* on this page, and the Book of Lore on pages 28-31 to learn the rules.

Going further

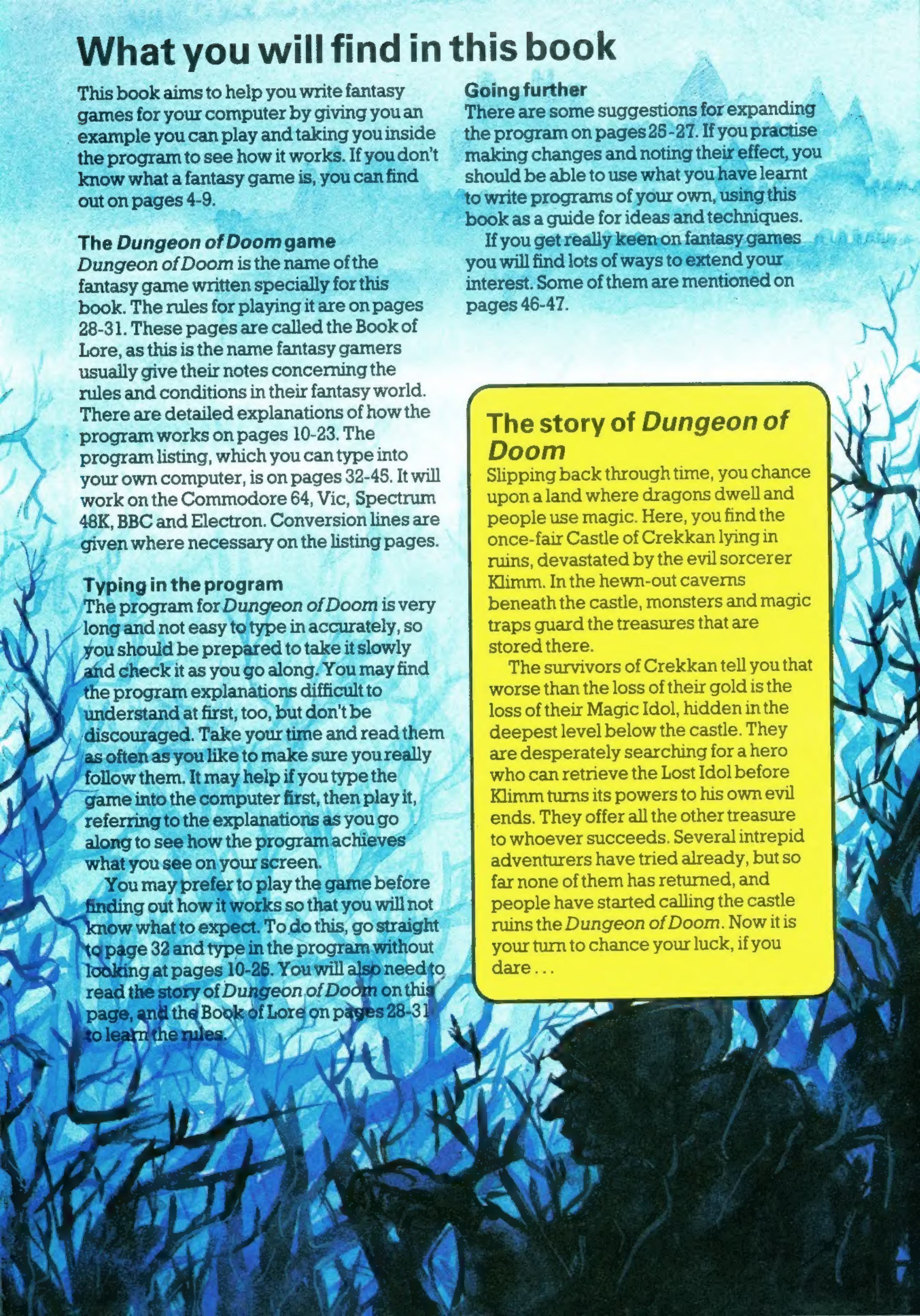
There are some suggestions for expanding the program on pages 25-27. If you practise making changes and noting their effect, you should be able to use what you have learnt to write programs of your own, using this book as a guide for ideas and techniques.

If you get really keen on fantasy games you will find lots of ways to extend your interest. Some of them are mentioned on pages 46-47.

The story of *Dungeon of Doom*

Slipping back through time, you chance upon a land where dragons dwell and people use magic. Here, you find the once-fair Castle of Crekkan lying in ruins, devastated by the evil sorcerer Klimm. In the hewn-out caverns beneath the castle, monsters and magic traps guard the treasures that are stored there.

The survivors of Crekkan tell you that worse than the loss of their gold is the loss of their Magic Idol, hidden in the deepest level below the castle. They are desperately searching for a hero who can retrieve the Lost Idol before Klimm turns its powers to his own evil ends. They offer all the other treasure to whoever succeeds. Several intrepid adventurers have tried already, but so far none of them has returned, and people have started calling the castle ruins the *Dungeon of Doom*. Now it is your turn to chance your luck, if you dare . . .

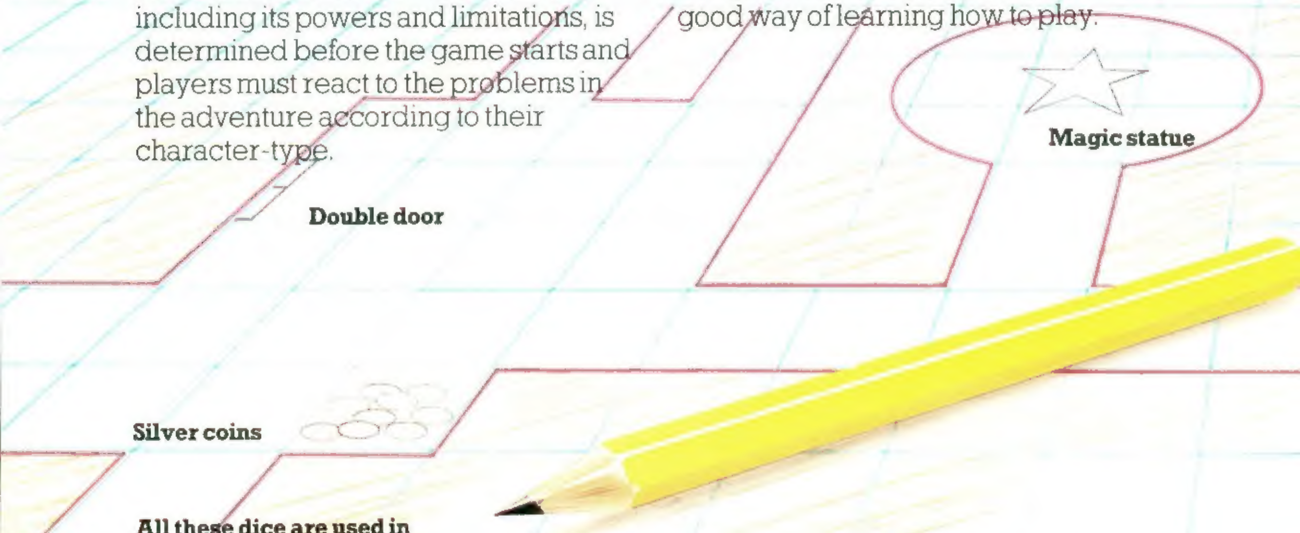


What is a fantasy game?

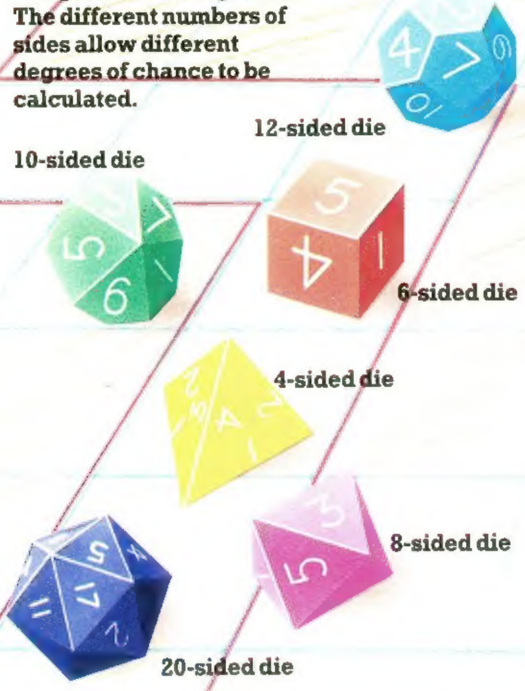
A fantasy game is an adventure game with a difference. An adventure game usually involves a dangerous expedition, or Quest, with obstacles to overcome, enemies to face and treasure to find. In fantasy games the added challenge is that players do not play as themselves but take on a character, or role. Because of this they are often called role-playing games, or RPGs. The nature of each character, including its powers and limitations, is determined before the game starts and players must react to the problems in the adventure according to their character-type.

The aim of the game is to gain experience and survive. A character is rewarded for success by being granted higher status, so a player can use the same hero in lots of adventures and his powers will develop all the time.

In a role-playing game, several players may take different character-types and go on a quest together, but a player can go on a solo quest, as in *Dungeon of Doom*, and this can be a good way of learning how to play.



All these dice are used in *Dungeons and Dragons. The different numbers of sides allow different degrees of chance to be calculated.**



*Dungeons and Dragons**

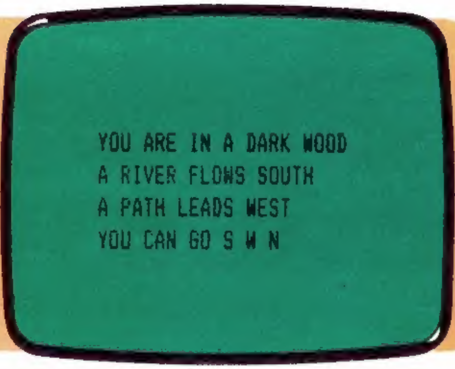
Anybody can invent a character and think of adventures for it, of course. But as a fantasy game gets more complicated, with lots of characters playing, and more monsters, magic and treasures, it is useful to have a set of rules as a guideline for conducting events. This ensures that the game is consistent and fair to all players.

The first people to publish the rules they had developed for their fantasy game were two Americans, Gary Gygax and Dave Arneson. They called their game *Dungeons and Dragons** and it appeared in 1974. In this game, you use many-sided dice, like the ones shown here, to decide character-types and calculate the outcome of battles. You draw a map as you explore, like the one that has been started on this page. You can buy *Dungeons and Dragons*, and many other fantasy games that have been invented since, in games shops (see pages 46-47).

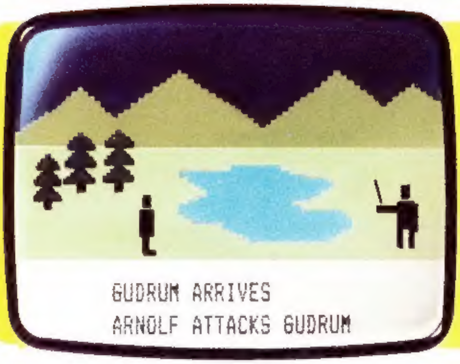
**Dungeons and Dragons is a registered trademark owned by TSR Hobbies Inc.*

Fantasy games for computers

As fantasy-type games became popular, people adapted them for computers. Games were developed on mainframes and were condensed for home computers. A micro has a small memory and is normally only used by one person at a time, so many programs are solo, text-only games, like the one on this screen. These are more like straight adventures than role-playing games.



```
YOU ARE IN A DARK WOOD
A RIVER FLOWS SOUTH
A PATH LEADS WEST
YOU CAN GO S W N
```

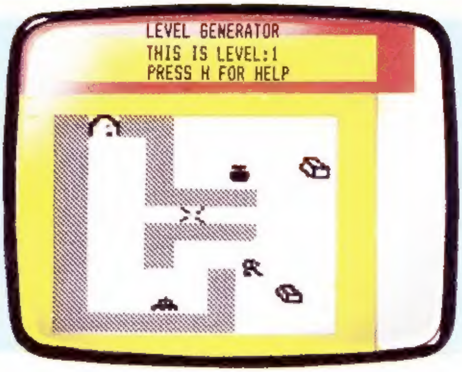


```
GUDRUN ARRIVES
ARNOLF ATTACKS GUDRUN
```

One solution is to use computers simply as an aid to fantasy games, and there are programs just for making characters, planning dungeons or deciding combat results. New games are appearing, though, which use graphics for locations, like the one shown here, or in which the solo player can take a different character each time he plays, or play with several characters at once.

Although *Dungeon of Doom* is a solo game, you can create a variety of characters for yourself, as in a non-computer role-playing game. You can also design dungeons, which appear as a plan on your screen. This screen shows a half-planned dungeon for *Dungeon of Doom**.

Some of the computer games you can buy are given on page 46.

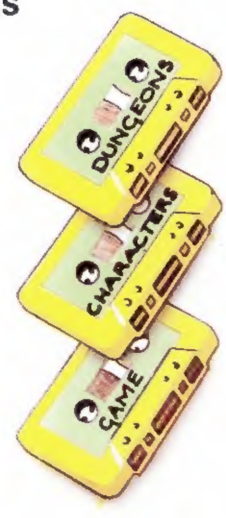


```
LEVEL GENERATOR
THIS IS LEVEL:1
PRESS H FOR HELP
```

What sort of program is *Dungeon of Doom?*

The program for *Dungeon of Doom* could be described as a "calculating database". This means that the program is based on numbers, and actions in the game are decided by calculations involving the relevant numbers. If this sounds complicated, don't worry; it is explained in detail later on.

The program uses sound effects for battles, and graphics which you program yourself (called "user-defined"). Once you understand how these work, you can adapt them for programs of your own.



The structure of the program could also be adapted. It is divided into three inter-relating parts. The routines you use to create dungeons and characters are programmed separately, then your creations are loaded into the game program when needed. You could use this structure in programs other than games where you want to work something out first and add the result to the main program later.

**Screen display may vary slightly on different computers.*

Dungeons and characters

To play a fantasy game, you first need a dungeon and a character who is armed and ready to enter it.

What is a dungeon?

A dungeon is where your fantasy game takes place. It is usually called a dungeon, but it doesn't actually have to be a dungeon at all. *Dungeon of Doom* is set beneath the ruins of the Castle of Crekkan, but you could venture to another planet where aliens are plotting the destruction of Earth, or into a cave where prehistoric creatures have survived. Even a city like London or New York could be the location for an exciting game.

To be really convincing, your dungeon should be filled with creatures and things that you might expect to find there. In a cave you might find a magic sword and evil trolls, but in Space, Martians with laser guns would be more likely.

Make your treasure appropriate, too. A Space adventurer could be searching for the Martians' plans to destroy the Earth, or a Knight of Old might be seeking buried gold.

Dungeon levels

Once you have collected the treasure from your dungeon and escaped alive, you can have further adventures in your fantasy world by inventing more dungeons, which should get progressively more difficult. Your character must gain experience in the easier dungeons, though, before he can try his luck in the more dangerous ones.

Here is a picture of some dungeons. There are three levels. The top one is the first one you enter and is the easiest to survive in. They get more dangerous as you go down but there are greater treasures, too. When you plan your dungeons it is up to you to grade them so that courage and experience are properly rewarded.

Inventing a character

A character is the hero who enters your dungeon. Like everything else, he should fit logically into the setting. A Knight of Old wouldn't get far on an alien planet.

Every character in a fantasy game has attributes, such as strength, agility, aura (magical ability) and so on. These are decided before the game starts. In *Dungeon of Doom* the computer gives you a score beside each attribute which shows how much of that quality your character has. You also get a few modification points with

Barbarian

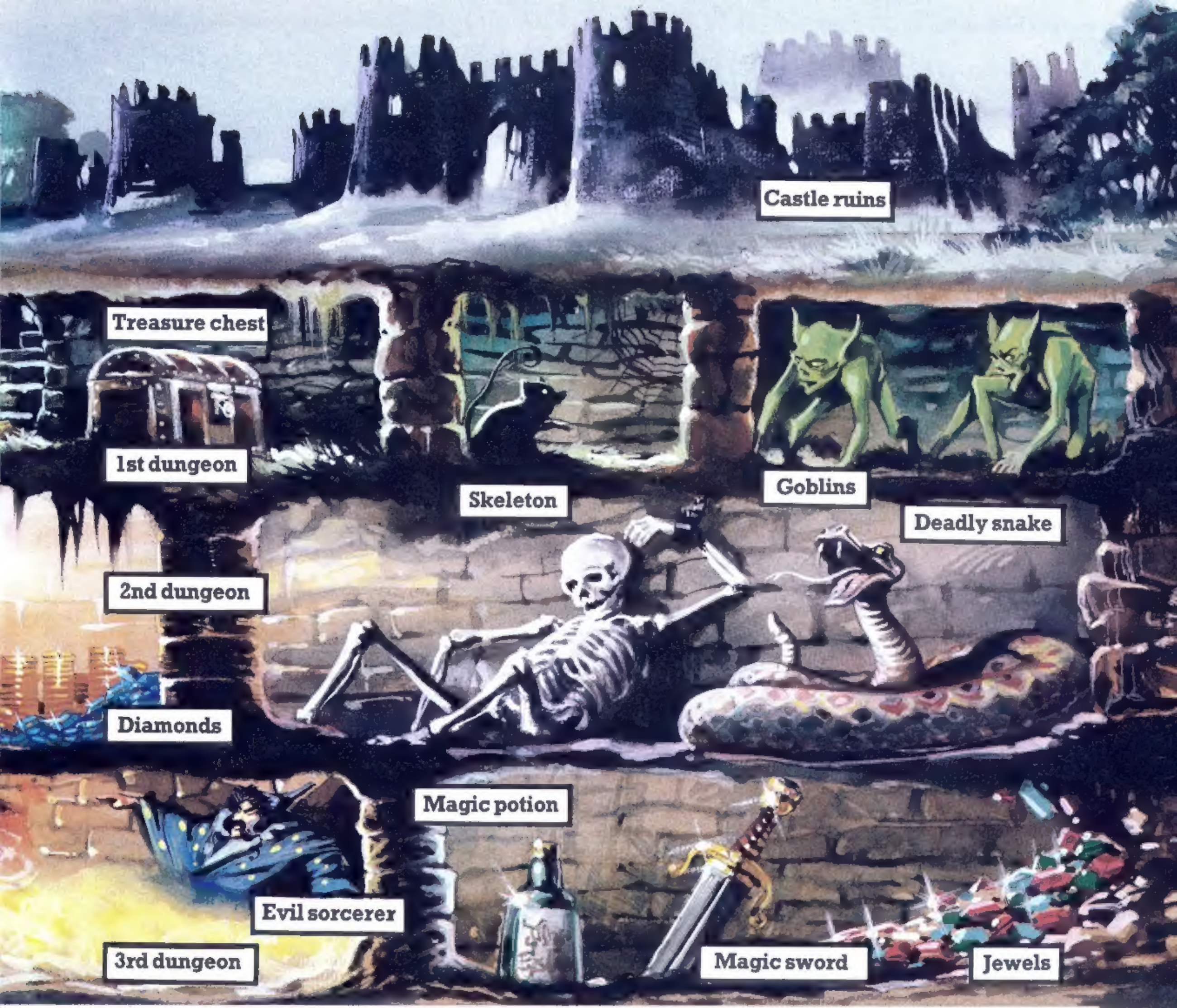
*Very strong and fierce.
Loves to fight.
Low intelligence.
No magic ability.*

Knight

*Also strong, but more intelligent. Does not always rush into a fight.
No magic.*

Scholar

*Less strong, but still a good fighter.
High intelligence.
Some magic ability.*



which you can change your scores if you want more intelligence, or less strength, for example .

The attribute scores determine your character-type. If you use your modification points to change them, then your character-type may change, too. (Making a character for *Dungeon of Doom* is fully explained on pages 18-19.)

These pictures show some possible character-types and their attributes for a typical dungeon setting. It's your job to make the best use of your character's skills.

Magician

Not strong, avoids physical combat. Very intelligent. Uses powerful magic.

Thief

Fairly strong and intelligent. Avoids combat. Prefers to use cunning where possible. No magic.

Dwarf

Tough and violent, despite being small. Has some lesser magic abilities, like seeing in the dark.'

Trading post and dungeon master

In most fantasy games, the character you create will have the chance to visit a shop or trading post to buy provisions before setting off on a quest. *Dungeon of Doom* follows this pattern and presents the player with a choice of armour, weapons, food, magic items and lights which the hero can buy. The computer shows how much gold the character has to start with and he can spend it as he likes.

Metal armour is strong but also heavy. Slows character down. Leather armour is lighter and more flexible.

Here are some weapons and armour your hero might carry.

Below are some of the magic items you might use.

Necronomicon (book of spells used by a necromancer, who calls up the dead).

Amulets (ornaments with magic signs to ward off evil).

Magic scrolls

Magic potions to heal wounds.

What different characters buy

In most fantasy games, the rules limit what each character-type can buy. This makes things fairer. A magician, for example, may be allowed to buy magic items but no heavy weapons, while a barbarian can buy arms and armour but no magic. This means that each character-type must fight according to his special skills.

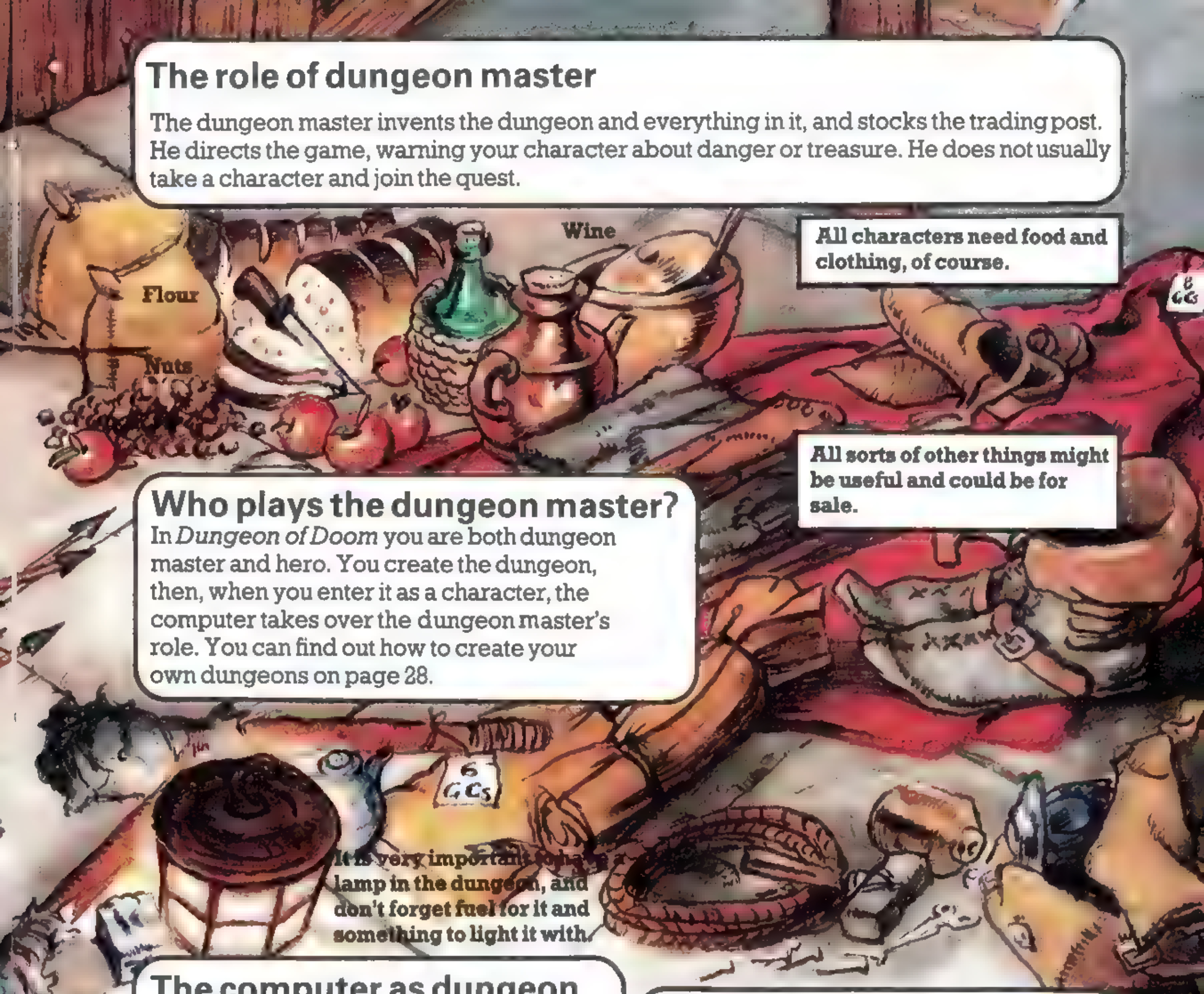
In *Dungeon of Doom* the computer refuses to let certain character-types buy certain items, so that none of them has an unfair advantage. When you create different dungeon settings and character-types you will probably find it necessary to have limitations like these, too.

Bargaining

It is quite normal to bargain for goods in the trading post. The computer allows the player to bargain in *Dungeon of Doom*, but may not accept his first offer, so he must keep trying until the price is agreed. The Book of Lore on page 29 tells you how you can spend your gold and bargain for goods in *Dungeon of Doom*.

The role of dungeon master

The dungeon master invents the dungeon and everything in it, and stocks the trading post. He directs the game, warning your character about danger or treasure. He does not usually take a character and join the quest.



All characters need food and clothing, of course.

All sorts of other things might be useful and could be for sale.

It is very important to have a lamp in the dungeon, and don't forget fuel for it and something to light it with.

Who plays the dungeon master?

In *Dungeon of Doom* you are both dungeon master and hero. You create the dungeon, then, when you enter it as a character, the computer takes over the dungeon master's role. You can find out how to create your own dungeons on page 28.

The computer as dungeon master

The computer keeps track of the state of play. It reveals the dungeon as your character explores, and changes his attributes as things happen to him. For example, his strength score will go down during a fight, and his experience score will increase. All the treasure he finds is recorded, too.

An unknown dungeon

You will probably not remember the exact location of objects in your dungeon, especially if you create more than one level at a time. However, if you want the dungeon to be a complete surprise to your hero, perhaps you could persuade someone else to design it for you. You and a friend could make dungeons for each other.

Practise being dungeon master

Try to think what the characters might need for a fantasy game in space, or in vampire country, or in a detective story setting? These shopping lists will give you a start.

Space Shop
Laser gun
Star map
Fuel rods

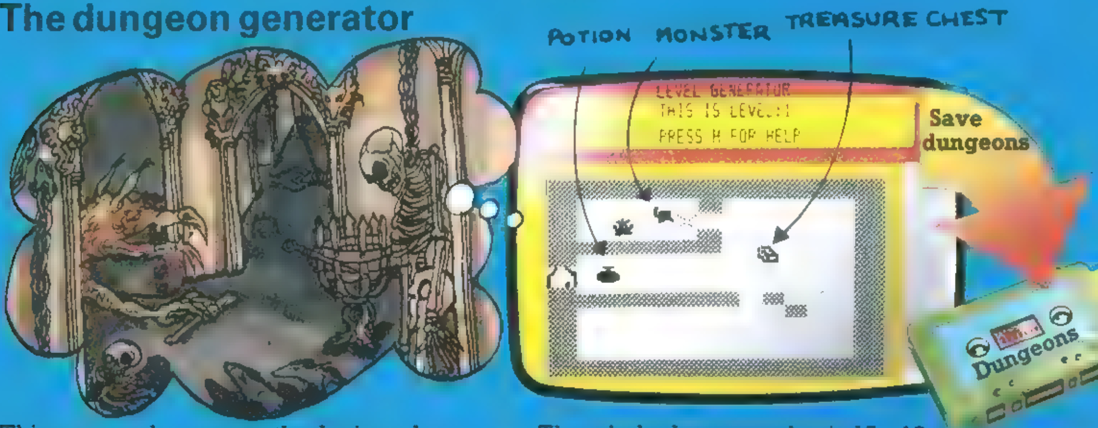
Dick's Store
Disguises
Bullets
Pistol

Transylvania Traders
Bible
Silver cross
Garlic

The *Dungeon of Doom* program

A good computer fantasy game program must let you make dungeons and characters, and must have exciting action routines, too. Each of these uses memory space and you might not have enough room if you try to put them all in one program, so *Dungeon of Doom* is divided into three shorter programs, called the dungeon generator, character creator and game module. Each one is kept on a separate cassette and you use them one at a time.

The dungeon generator

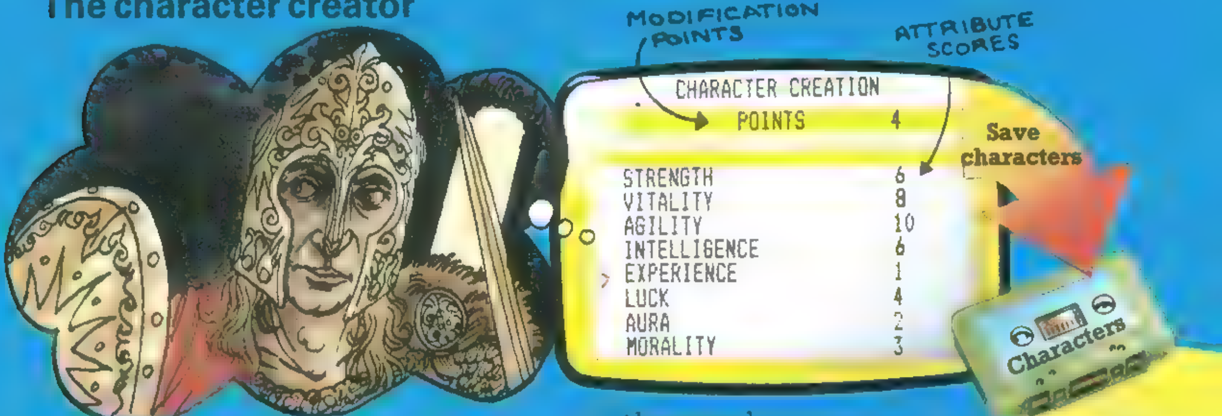


```
LEVEL GENERATOR
THIS IS LEVEL:1
PRESS H FOR HELP
```

This screen shows a partly-designed dungeon, made using the dungeon generator program.* This allows you to place objects on the dungeon plan using graphic symbols. You move the cursor into position, then place an object there by pressing a number key. Each symbol occupies a letter-sized space on the screen.

The whole dungeon plan is 15 × 15 spaces. This leaves enough room on the screen for the rest of the display (messages and attribute scores) and does not use up too much memory when fed into the game module. Once you have typed the program into your computer you can save it on a cassette. Then save the dungeons you plan on the other side.

The character creator



```
CHARACTER CREATION
     POINTS           4

STRENGTH              6
VITALITY              8
AGILITY               10
INTELLIGENCE          6
EXPERIENCE            1
LUCK                  4
AURA                  2
MORALITY              3
```

The screen above shows a character being made using the character creator.* This program is in four "pages", which appear on the screen one at a time. The first page shows attributes and modification points for changing them. Any modifications you make may change the character-type, which appears on the screen. You cannot modify the experience score; more points must be earned.

The other three pages offer goods for sale and gold coins to spend. There are limitations on what items different character-types can buy, and they can all be bargained for. Finally, you can name your character, using up to ten letters.

You can save the character creator on one side of a cassette and any characters you make on the other.

**The screen displays may vary slightly from these on different computers.*

The game module

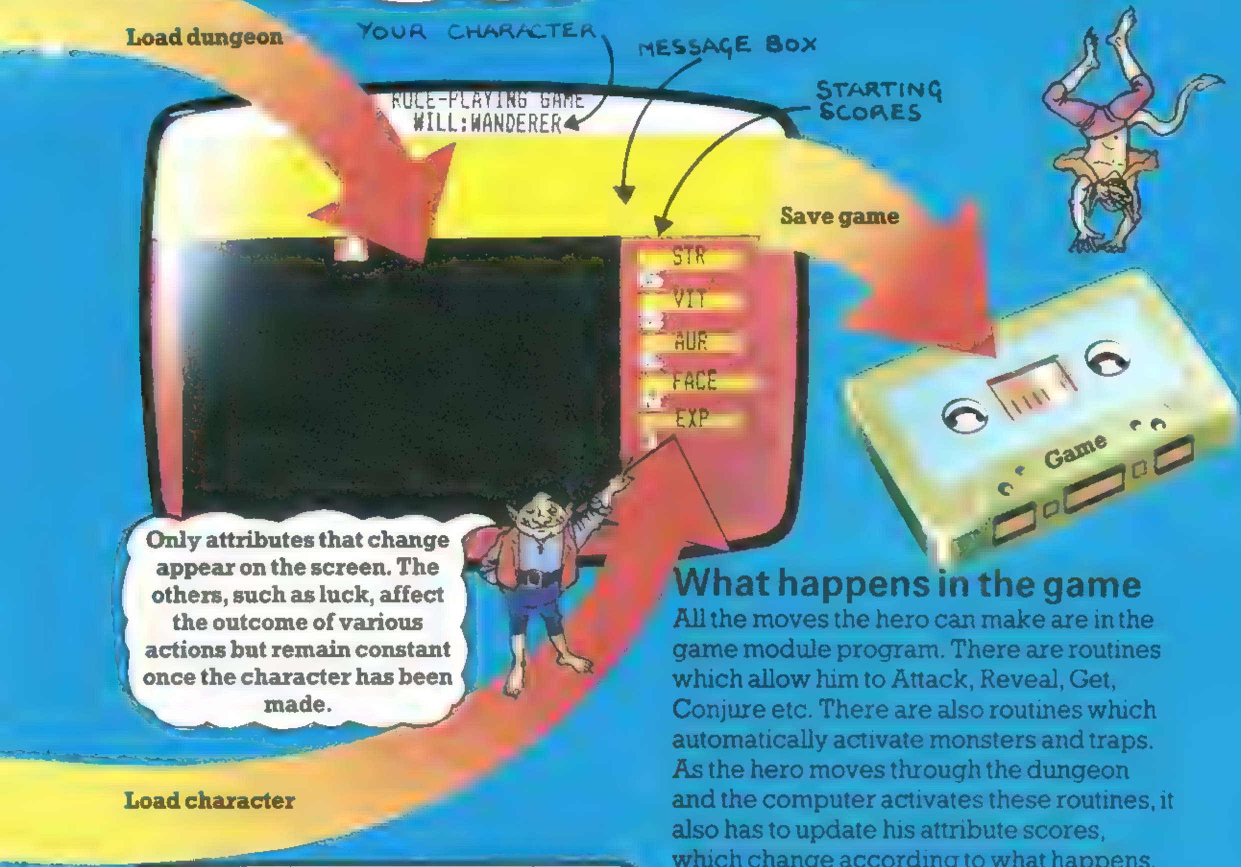
The game module is the program you use when you are ready to play the game. It asks you to load a hero and a dungeon from their cassettes, so you must design at least one of each beforehand. You load the hero first and the computer looks at his experience score. Then when you load a dungeon it compares the dungeon level number with the hero's experience. It will not allow a character to enter a dungeon level that is too advanced.

The game module must contain data to interpret the dungeon and character that are loaded, but does not need to know how they were made so the data doesn't take up much memory space.

What you see on your screen

The game module sets up your screen as shown below.* The graphic symbol that represents your character is placed in the entrance to the dungeon you have loaded. The attributes are displayed on the right of the screen. The rest of the dungeon is in darkness, waiting for the hero to explore.

The computer numbers your dungeons so you can save them one after the other, with easier ones first and more difficult ones later. The object of the Quest (the Lost Idol) should be in the last dungeon.



Only attributes that change appear on the screen. The others, such as luck, affect the outcome of various actions but remain constant once the character has been made.

What happens in the game

All the moves the hero can make are in the game module program. There are routines which allow him to Attack, Reveal, Get, Conjure etc. There are also routines which automatically activate monsters and traps. As the hero moves through the dungeon and the computer activates these routines, it also has to update his attribute scores, which change according to what happens.

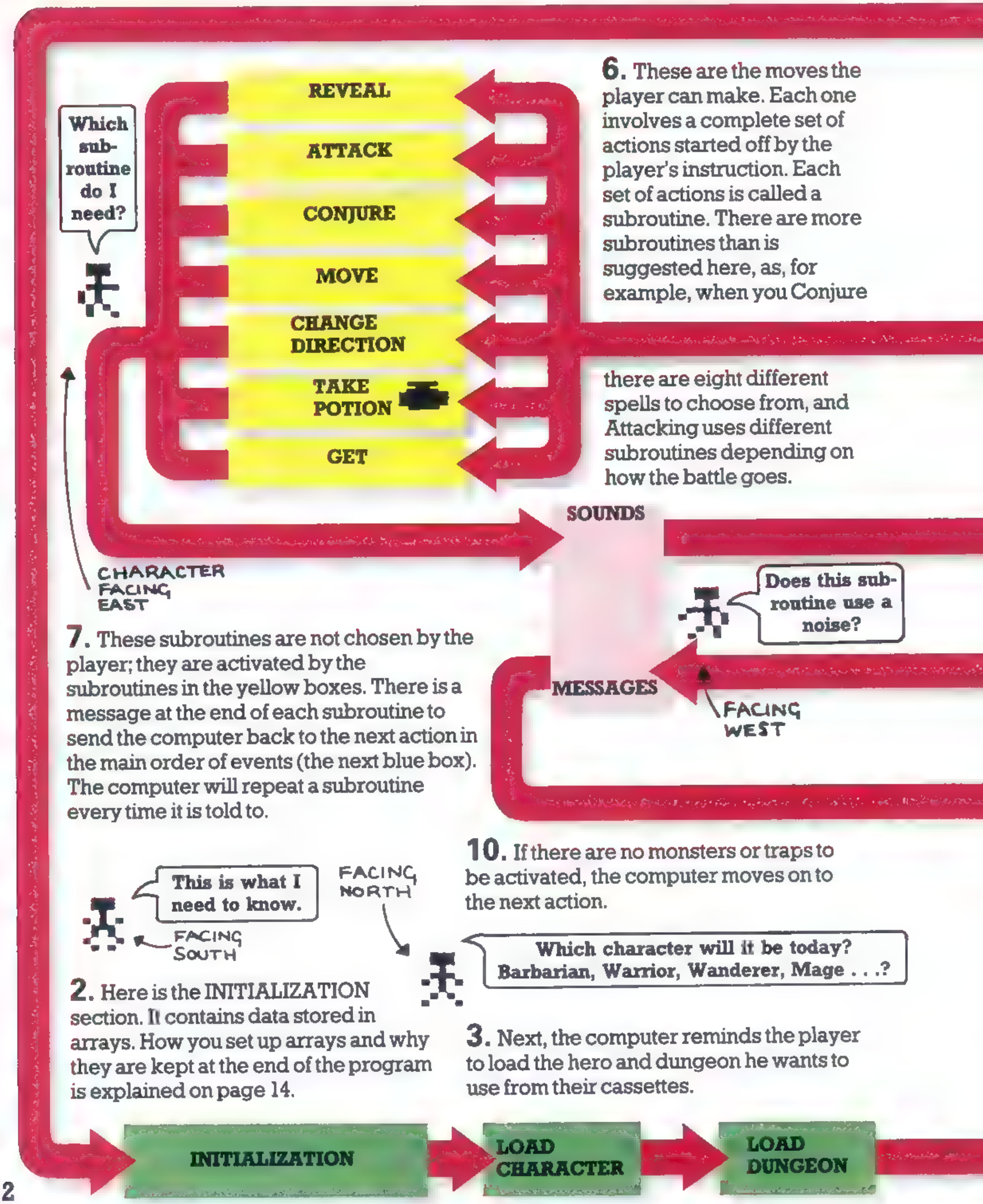
Saving the game

You can save the game module program on one side of a third cassette. You can then save a game in the middle of play on the other side. The attribute scores and the hero's position in the dungeon will be saved so you can start again where you left off.

By using three cassettes like this, you can make or erase dungeons and characters as you like and save a current game without accidentally erasing part of the program. Remember to label your cassettes clearly.

What the game module does

These two pages show you the structure of the game module program. The actions in the blue boxes show the main order of events. If you follow the red line from START you will discover how the computer works through the program and does all the jobs necessary to play *Dungeon of Doom*.


REVEAL
ATTACK
CONJURE
MOVE
CHANGE DIRECTION
TAKE POTION
GET
Which sub-routine do I need?
CHARACTER FACING EAST
SOUNDS
Does this sub-routine use a noise?
MESSAGES
FACING WEST
This is what I need to know.
FACING SOUTH
FACING NORTH
Which character will it be today? Barbarian, Warrior, Wanderer, Mage . . .?
INITIALIZATION
LOAD CHARACTER
LOAD DUNGEON

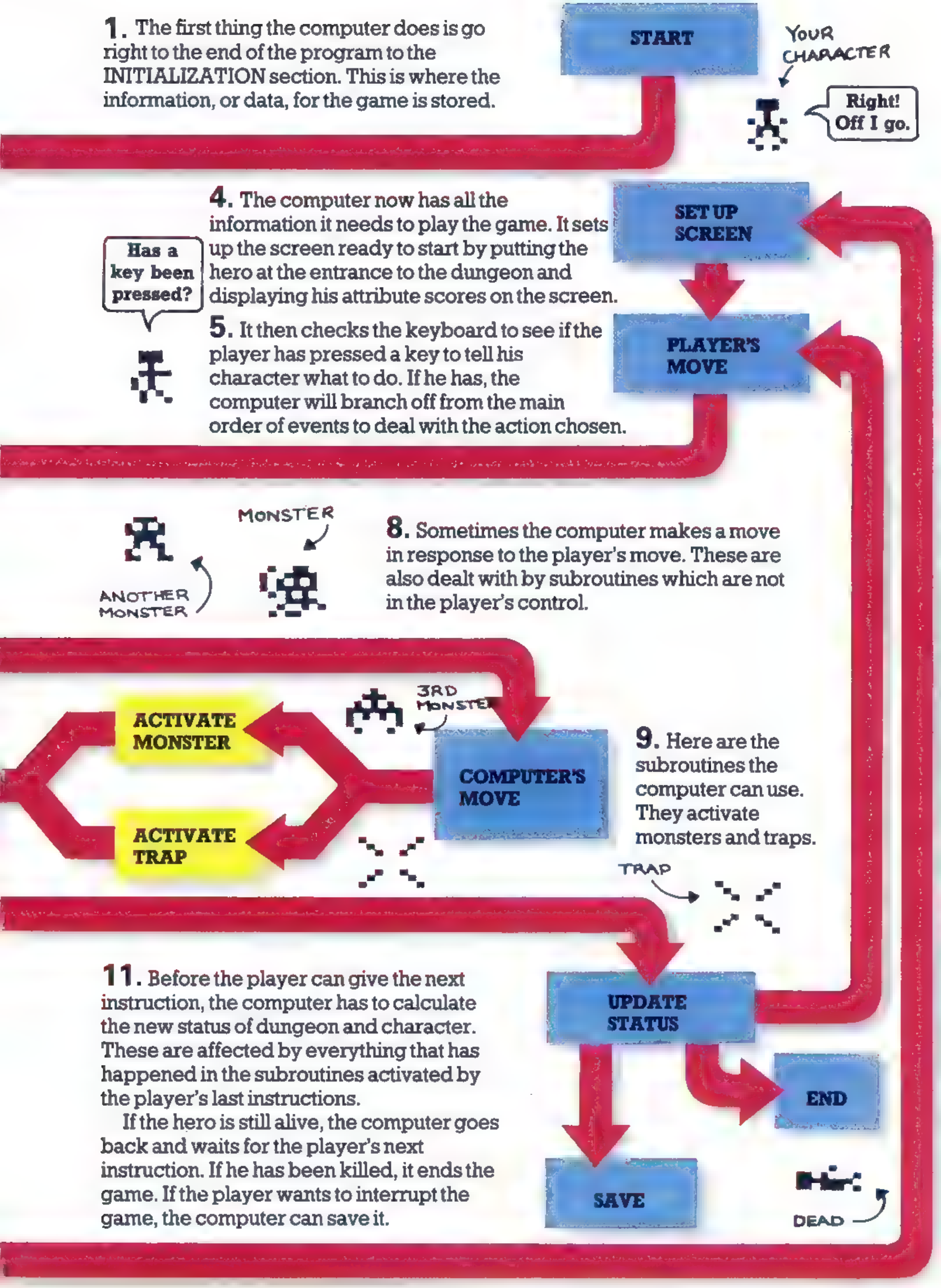
2. Here is the INITIALIZATION section. It contains data stored in arrays. How you set up arrays and why they are kept at the end of the program is explained on page 14.

3. Next, the computer reminds the player to load the hero and dungeon he wants to use from their cassettes.

6. These are the moves the player can make. Each one involves a complete set of actions started off by the player's instruction. Each set of actions is called a subroutine. There are more subroutines than is suggested here, as, for example, when you Conjure there are eight different spells to choose from, and Attacking uses different subroutines depending on how the battle goes.

7. These subroutines are not chosen by the player; they are activated by the subroutines in the yellow boxes. There is a message at the end of each subroutine to send the computer back to the next action in the main order of events (the next blue box). The computer will repeat a subroutine every time it is told to.

10. If there are no monsters or traps to be activated, the computer moves on to the next action.



1. The first thing the computer does is go right to the end of the program to the INITIALIZATION section. This is where the information, or data, for the game is stored.

4. The computer now has all the information it needs to play the game. It sets up the screen ready to start by putting the hero at the entrance to the dungeon and displaying his attribute scores on the screen.

5. It then checks the keyboard to see if the player has pressed a key to tell his character what to do. If he has, the computer will branch off from the main order of events to deal with the action chosen.

8. Sometimes the computer makes a move in response to the player's move. These are also dealt with by subroutines which are not in the player's control.

9. Here are the subroutines the computer can use. They activate monsters and traps.

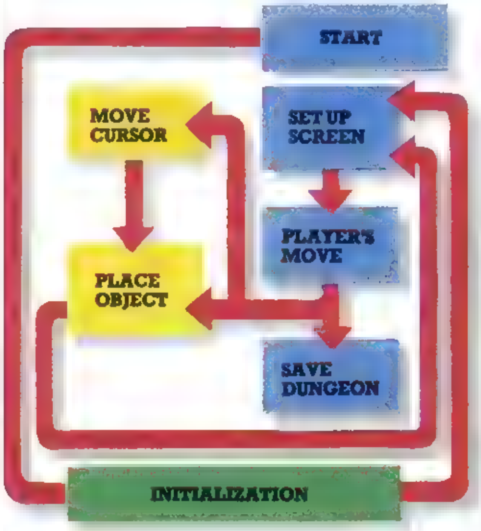
11. Before the player can give the next instruction, the computer has to calculate the new status of dungeon and character. These are affected by everything that has happened in the subroutines activated by the player's last instructions.

If the hero is still alive, the computer goes back and waits for the player's next instruction. If he has been killed, it ends the game. If the player wants to interrupt the game, the computer can save it.

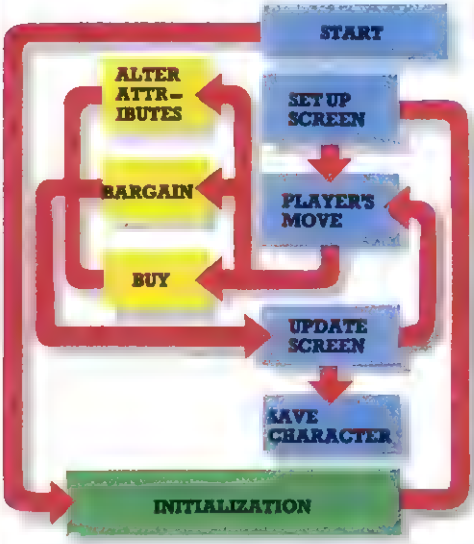
What the dungeon generator and character creator do

On the last two pages you saw how the computer works through the game module program. The dungeon generator and character creator programs have a similar structure, although they are shorter as they have fewer jobs to do.

This is what the dungeon generator program looks like:



And here is what the character creator program looks like:



If you compare these with the game module, you will see the similarities.

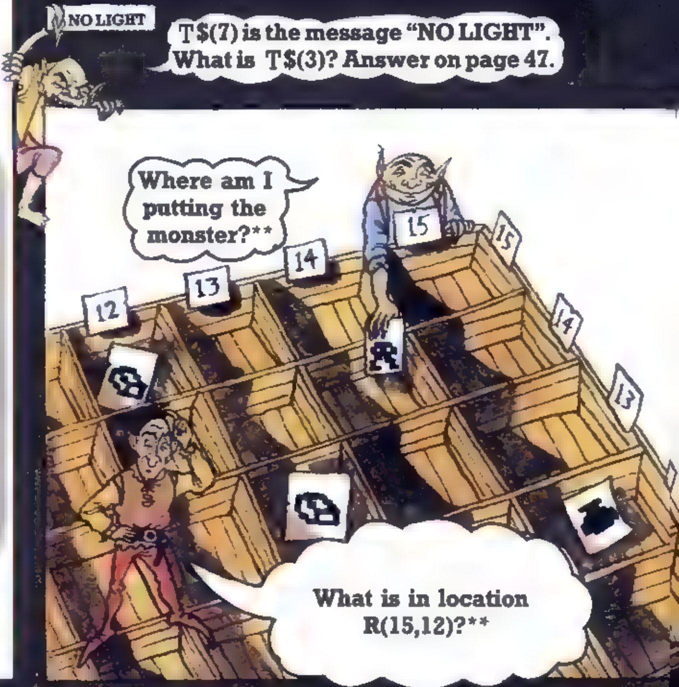
Storing the data

Here, and on the next nine pages, you will find explanations of how the three parts of the program work. On this page you can find out about initialization. Each part of *Dungeon of Doom* has its own initialization section, and it is this which contains the data the computer needs to run the program.

To put data into the computer you have to tell it to put aside space in its memory. This space is called an array and it is like a row of boxes. An array for *Dungeon of Doom* might look like this.

2
1
T$
Your array must have a name so it can be found again easily.

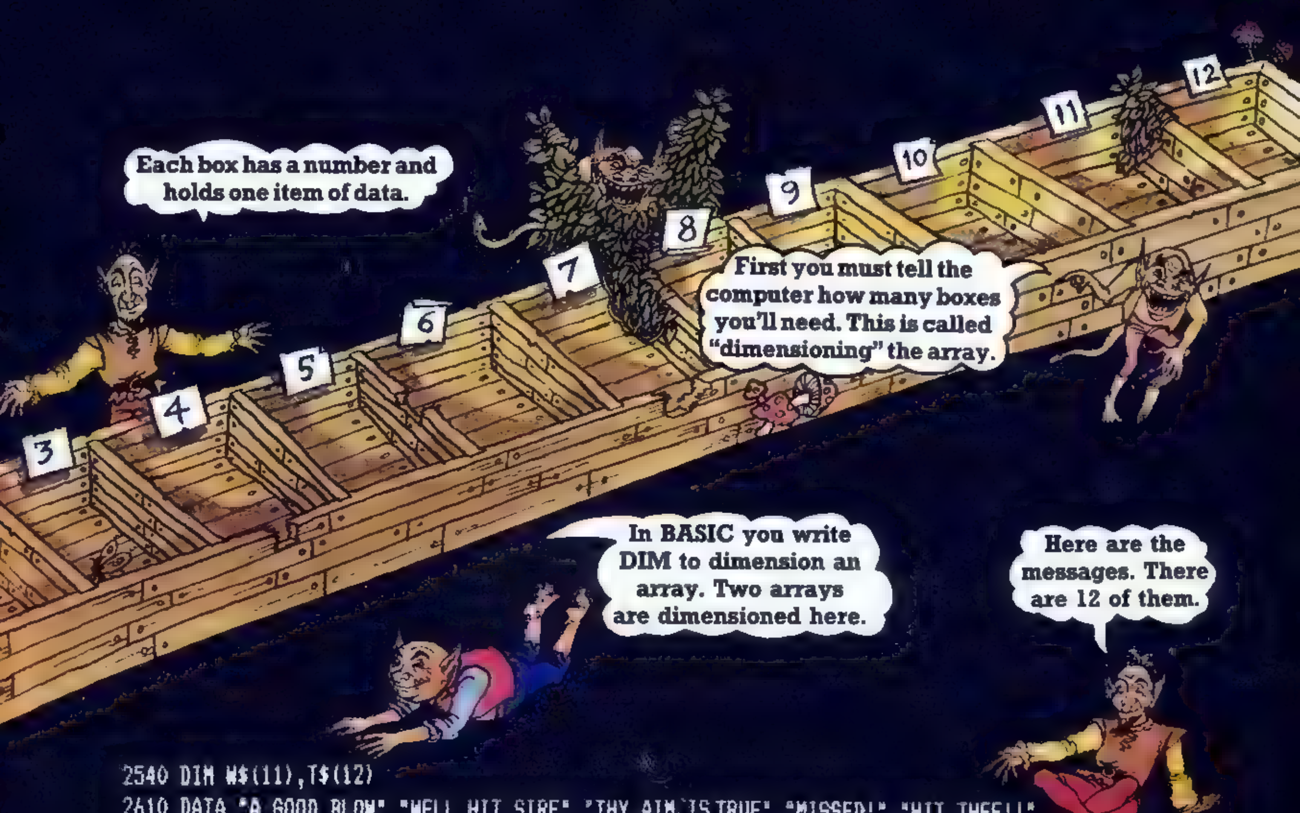
The array below is called T$ (you say "T string", or "T dollar"). It contains the messages to be used in *Dungeon of Doom*. It has to have the sign $ after its name because the data in it is words. It is called a string array. A numbers-only array can be labelled without the $. These lines from the game module show how to set up this array.*


NO LIGHT
T$(7) is the message "NO LIGHT". What is T$(3)? Answer on page 47.
Where am I putting the monster?**
15
14
13
12
15
14
13
What is in location R(15,12)?**

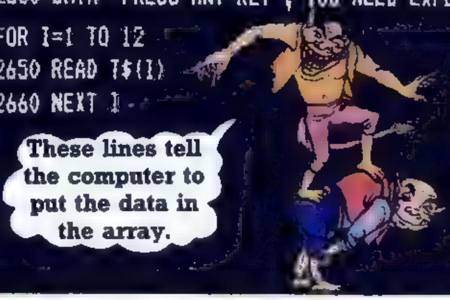
---

*Machine specific conversions start on page 45.

**Answers on page 47


Each box has a number and holds one item of data.
3
4
5
6
7
8
9
10
11
12
First you must tell the computer how many boxes you'll need. This is called "dimensioning" the array.
In BASIC you write DIM to dimension an array. Two arrays are dimensioned here.
Here are the messages. There are 12 of them.

```
2540 DIM W$(11),T$(12)
2610 DATA "A GOOD BLOW","WELL HIT SIRE","THY AIM IS TRUE","MISSED!","HIT THEE!!"
2620 DATA "THE MONSTER IS SLAIN","NO LIGHT","BROKEN THY ","SPELL EXHAUSTED"
2630 DATA "PRESS ANY KEY","YOU NEED EXPERIENCE","EXIT FROM THIS LEVEL"
FOR I=1 TO 12
2650 READ T$(I)
2660 NEXT I
```


These lines tell the computer to put the data in the array.

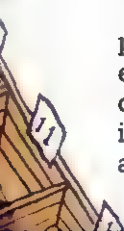
To put the data in the array the computer loops round 12 times and puts one item of data from the data list into the next space in the array each time.

Whenever a message is needed during the game it will be referred to like this: T$(5), where the number in brackets tells the computer which box to look in.

2D arrays

T$ is a one-dimensional (1D) array. *Dungeon of Doom* also uses two-dimensional (2D) arrays. A 2D array is like rows of boxes stuck together in a unit (see picture opposite). This unit stores the room-contents of the dungeons you plan. It is called R and contains 15 × 15 boxes. You dimension it like this: DIM R(15,15). A box in the unit is referred to like this: R(3,2). The first number is the column it is in, and the second is the row.

The character creator uses a 2D array for the four pages of attributes and possessions. Each page holds eight items. To refer to any individual item, you tell the computer which page it is on and its position in the list. F is the label for attribute scores, so F(1,4) is the fourth attribute on the first page.



Why initialization is at the end

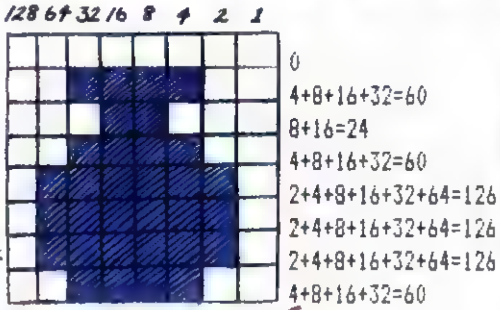
Although the computer reads the initialization section first, it is placed at the end of each of these programs. Initialization is a large block of data, occupying many lines. If it were at the start it would slow the game down, because, whenever the computer is sent to find a subroutine, it checks every line number from the beginning until it reaches the right one.

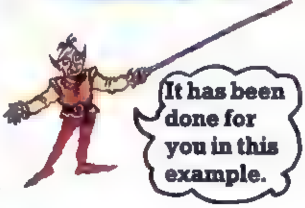
How the dungeon generator works

In the dungeon generator program you use graphic symbols that are not standard in your computer's memory – monsters, potions, treasures and so on. These are called "user-defined graphics" and are programmed in the initialization section.

The example below shows how the potion symbol was designed and programmed. The same method is used whatever micro you have, although the commands may differ.

1. On a piece of paper draw a grid of 8×8 squares to represent the space your symbol will occupy. Your whole screen is made up of tiny squares and all keyboards symbols are defined in an area of 8×8 squares so your new symbol will be, too.
2. Shade in the squares to make the shape you want.
3. Define the symbol for your computer row by row. To do this, you number the columns as in the picture. Then give a value to each row by adding together the numbers at the top of each column where there is a shaded square.




It has been done for you in this example.

```
1030 DATA 0,60,24,60,126,126,126,60
```

This line describes your new graphic symbol to the computer.

The way you feed this data into your computer varies enormously. Space has to be found for it in the memory, where everything is referred to by code.(Most micros use the same code, called ASCII – American Standard Code for Information Interchange). Many of the available code numbers are used to refer to the keyboard symbols and other data the computer needs to work. But most computers have some "spare" code numbers that can be used to label new data put in by you.

Putting symbols on the screen

You place symbols on the screen by pressing a number key. The computer needs instructions for relating the key you press to the ASCII code for a particular symbol. Here are the lines that do this:

```
230 LET I=VAL(I$)
```

A number is the same as any other keyboard symbol to the computer and is referred to by an ASCII code. VAL tells the computer not to refer to the code for that key, but to its face value.

```
260 LET R(X,Y)=C0+I
```

R(X,Y)=location C0=OS+6 I=key pressed

OS is an offset number, used to calculate ASCII codes for graphic symbols. Here, OS=ASCII code for the first symbol. The next one is OS+1, and so on. However, the first five symbols are only used in the game module (see page 44) and do not relate to number keys, so a definition C0 is introduced. C0=OS+6 (see line 770). The computer now skips the first five symbols and calculates C0+I to find the correct code.

More routines in the dungeon generator

The dungeon generator's job is to let you position the symbols you have defined on your dungeon plan. Here are the commands that allow you to do so:

H Sends the computer to the Help routine, which reminds you which number represents which symbol.

A Moves cursor up screen.

S Saves dungeon (see below).

```
110 IF I$="H" THEN GOSUB 360
120 IF I$="A" AND Y>1 THEN LET Y=Y-1
130 IF I$="Z" AND Y<15 THEN LET Y=Y+1
140 IF I$="N" AND X>1 THEN LET X=X-1
150 IF I$="M" AND X<15 THEN LET X=X+1
160 IF I$>"/" AND I$<":" THEN GOSUB 230
```

Z Moves cursor down screen.

M Moves cursor to the right.

N Moves cursor to the left.

5 The symbols that come between "/" and ":" in the computer's memory are numbers.

Saving the dungeon

Finally, the dungeon generator must allow you to save your dungeon. It does this by remembering which symbol is in each location (a blank counts as a symbol, too) and saving this information in a long line as a string variable, called S$. S$ can then be loaded into the game module. Here you can see the save and load dungeon routines side by side.

DUNGEON GENERATOR

```
460 LET S$=""
470 FOR J=1 TO 15
480 FOR K=1 TO 15
490 LET S$=S$+CHR$(R(K,J))
500 NEXT K
510 NEXT J
520 LET S$=S$+CHR$(IX+OS):
    LET S$=S$+CHR$(IY+OS)
530 LET S$=S$+CHR$(LE+OS)
540 PRINT tab(1,4)"ANY KEY TO SAVE";:
    GOSUB 430
550 OPEN 1,1,1,"LEVEL"
560 PRINT£1,S$
570 CLOSE 1
580 PRINT tab(1,4);
    LEFT$(B$,W)
590 LET LE=LE+1:GOSUB 700
600 RETURN
```

S$ is loaded into game.

Dimensions of dungeon appear in both programs.

CHR$ tells computer that the ASCII code in location R(X,Y) represents a symbol. ASC "decodes" CHR$ in the game module.*

Location of entrance is saved by adding OS, and loaded into the game by subtracting OS.

Level number is also saved and loaded by adding, then subtracting OS.

This makes sure the level number increases.

GAME MODULE

```
1770 PRINT tab(0,3);"PREPARE DUNGEON TAPE"
1780 LET M$=T$(10):GOSUB 370
1790 OPEN 1,1,0,"LEVEL"
1800 INPUT£1,S$
1810 CLOSE 1
1820 I=1
1830 FOR Y=1 TO 15
1840 FOR X=1 TO 15
1850 LET R(X,Y)
=ASC(MID$(S$,I,1))
1860 LET I=I+1
1870 NEXT X
1880 NEXT Y
1890 LET IX=ASC(MID$(S$,I,1))-OS
1900 LET IY=ASC(MID$(S$,I+1,1))-OS
1910 LET LE=ASC(MID$
(S$,I+2,1))-OS
1920 IF LE>F(5) THEN GOSUB 1960:GOTO 1770
1930 GOSUB 2790
1940 LET NX=IX:LET NY=IY:LET OX=NX:LET OY=NY:
     LET DX=255
1950 RETURN
```

MID$ tells computer to pick a symbol out of the middle of S$.*

Checks level number against character's experience score (F = array of attributes – see 2D arrays on page 14).

**Not on the Spectrum. Conversions are given where these commands occur.*

How the character creator works

These pages look at the character creator. Initialization is at the end of the listing as usual. Have a look at the arrays. It is easy to pick out F$, which names the character's attributes and goods for sale as it is a list of words. But there are two number arrays which you may find confusing. These are explained below.

```
1240 DATA 20,16,12,15,8,10,8,6
1250 DATA 18,15,9,9,14,8,6,6
1260 DATA 20,15,14,12,10,8,6,6
```

This array is called F. It contains the prices of goods on sale.
Remember F is a number array and F$ is a word array. They are two completely different things.

```
1320 DATA 5,4,3,3,2,2,1,1
1330 DATA 5,4,3,1,2,1,3,1
1340 DATA 4,3,2,2,3,1,1,1
```

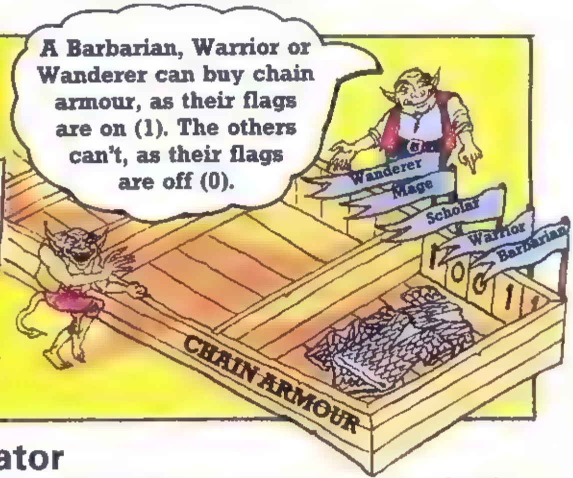
This one is called P (for "protection"). These numbers show how much offensive power (magic, weapons) or defensive power (armour, potions) the goods in F$ have. The computer uses these values in action routines in the game module.

Flags

Here is another number array which looks rather different.

```
1140 DATA"00001","00011","10011".......
1150 DATA"00011","00011","10011".......
1160 DATA"11100","00100","11100".......
```

These sets of 1s and 0s are called flags. For each of the goods in F$ there are five flags, which indicate whether a certain character can buy that item or not.


A Barbarian, Warrior or Wanderer can buy chain armour, as their flags are on (1). The others can't, as their flags are off (0).
Wanderer
Mage
Scholar
Warrior
Barbarian
1
0
0
1
1
CHAIN ARMOUR

Routines in the character creator

The lines below define the character-type according to changes in important attribute scores. If you look in F$ and C$ to find the words you need you can "translate" these lines into sentences:

120 If intelligence score is more than 6 and morality score is more than 7, then character-type will be a cleric.
See if you can write similar sentences for the other three conditional lines.*

To find out which attributes are concerned, read F$. The section on 2D arrays on page 15 tells you how.

Character-types are named in C$.

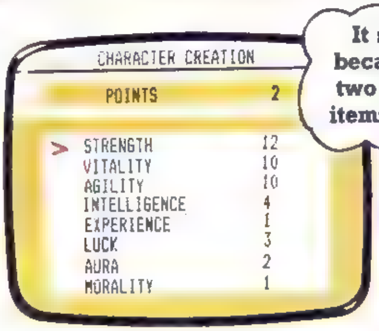
```
120 IF F(1,4)>6 AND F(1,8)>7 THEN LET C=2
130 IF F(1,4)>8 AND F(1,7)>7 THEN LET C=3
140 IF F(1,1)>7 AND F(1,8)>5 AND F(1,1)+F(1,2)>10 THEN LET C=4
150 IF F(1,1)>8 AND F(1,2)+F(1,3)>12 AND F(1,8)<6 THEN LET C=5
```

Finding items on the lists

The character creator is in four pages with eight items on each. The line which tells the computer which item you want to modify or buy looks like this:

```
270 LET N=8*(J-2)+K
```

N is the item you indicate with the cursor.
J is the number of rows used on each page.
K tells the computer how many rows to go down the page to find the item.
8 tells it that there are 8 items on each page.



```
CHARACTER CREATION

POINTS          2

> STRENGTH      12
  VITALITY      10
  AGILITY       10
  INTELLIGENCE   4
  EXPERIENCE     1
  LUCK           3
  AURA           2
  MORALITY       1
```

It says (J−2) because the top two lines aren't items on the list.

---

*Answers on page 47.

Bargaining

Bargaining is an important part of the program. Here's how this routine works.

You type in the amount you offer.

Subroutine 860 tells computer where to display message.

This subroutine checks if flag is on or off, to tell the computer if the character can buy this item.

This prevents you buying more than one of each item, except potions and salves (objects 23 and 24).

This calculates the bargain price (which you are not told). BR= a random number between 1 and 3.

H=gold coins left. If it's not enough to pay the original price of the item, you can't buy it or even bargain for it.

If offer is equal to or more than bargain price, you can have item. If not, you can't.

```
570 LET M$="":GOSUB 860
580 PRINT tab (2,2);"YOUR OFFER";
590 INPUT OF
600 GOSUB 680
610 IF O(N)>0 AND N<23 THEN LET M$=
    "YOU HAVE IT SIRE":RETURN
620 LET PR=F(J,K)-BR
630 IF H<PR THEN LET M$="YOU CANNOT AFFORD":
    RETURN
640 IF OF>PR AND Y=1 THEN LET O(N)=O(N)+P(N):
    LET H=H-PR:LET M$="TIS YOURS!"
650 IF OF<PR AND Y=1 THEN LET M$="OFFER REJECTED"
660 IF H<0 THEN LET H=0
670 RETURN
```

Saving the character

Your character is saved in a similar way to the dungeon (see page 17). The data in this case is not graphic symbols, but numbers. However, to avoid the numbers being confused with ASCII codes already in use in the game module, they are saved as if they were symbols by adding an offset number, AS.

CHARACTER CREATOR

```
420 LET O=O*3
430 LET S$=CHR$(O+AS)
440 FOR I=1 TO 8
450 LET S$=S$+CHR$(F(1,I)+AS)
460 NEXT I
470 FOR I=1 TO O
480 LET S$=S$+CHR$(O(I)+AS)
490 NEXT I
500 LET S$=S$+CHR$(H+AS)
510 LET S$=S$+CHR$(AS)
520 LET S$=S$+N$+" :"+C$(C)
530 OPEN 1,1,1,"HERO"
540 PRINT£1,S$
550 CLOSE 1
560 STOP
```

Number of objects.

AS added to value of object (O) to save it.

The first eight numbers are attribute scores.

These are the values of the objects.

C$(C) = name and type of character.

Starting values (S1, S2, and S3) are recorded for strength, vitality, and experience.

GAME MODULE

```
2030 OPEN 1,1,0,"HERO"
2040 INPUT£1,S$
2050 CLOSE 1
2060 LET P=2
2070 LET OT=ASC(MID$(S$,1,1))-AS
2080 FOR I=1 TO 8
2090 LET F(I)=ASC(MID$(S$,P,1))-AS
2100 LET P=P+1
2110 NEXT I
2120 FOR I=1 TO OT
2130 LET O(I)=ASC(MID$(S$,P,1))-AS
2140 LET P=P+1
2150 NEXT I
2160 LET GC=ASC(MID$(S$,P,1))-AS
2170 LET TR=ASC(MID$(S$,P+1,1))-AS
2180 LET C$=RIGHT$(S$,LEN(S$)-(P+1))
2190 LET S1=F(1):LET S2=F(2):LET S3=F(5)
2200 FOR I=1 TO 2
2210 FOR J=1 TO 3
2220 LET M((I-1)*3+J)=O(16+I)*F(7)
2230 NEXT J:NEXT I
2240 IF O(16)=1 THEN LET LT=20
2250 RETURN
```

Loads S$

AS subtracted and CHR$ decoded by command ASC*

Moves one row down to print next number.

GC=gold coins.

TR=treasure collected.

M=magic available to character.

*Not on Spectrum. Conversions are given beside listing, where necessary.

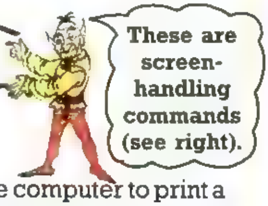
How the game module does its job

The game module deals with the hero's encounters with objects in the dungeon. The graphic symbols for these objects are defined as C0-C8, so you can specify certain symbols, or groups of symbols. C7 is the last object before the monsters, for example, so you can say that objects with a value of more than C7 should behave like monsters.

Service subroutines

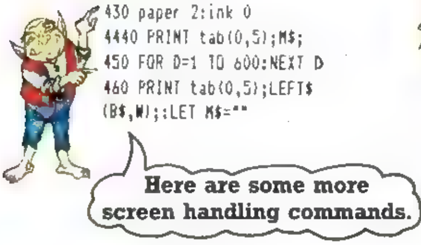
These subroutines are shown in the pink box in the picture on pages 12-13. They are routines for messages and sounds which are frequently activated by the action subroutines (in the yellow boxes).

```
380 PRINT tab(0,5);M$
390 LET I$=inkey$
400 IF I$="" THEN GOTO 390
410 PRINT tab(0,5);LEFT$
(B$,W);:LET M$=""
```


These are screen-handling commands (see right).

The one above tells the computer to print a message, then wait until a key is pressed. It is used in the conjure routine when you have to choose a spell. The one below prints a message on your screen for a short time.

```
430 paper 2:ink 0
4440 PRINT tab(0,5);M$;
450 FOR D=1 TO 600:NEXT D
460 PRINT tab(0,5);LEFT$
(B$,W);:LET M$=""
```


Here are some more screen handling commands.

Message stays on your screen while computer counts to 600.

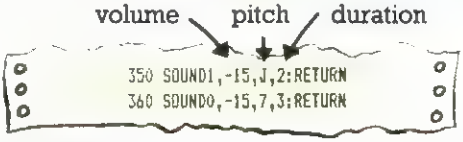
Screen-handling

In both message subroutines (see left) there are screen-handling commands to tell the computer any colour changes you want to make and where to print on the screen. Different computers have different colour commands. The Commodore 64 has special colour keys. The Spectrum uses PAPER (background) and INK (foreground) plus a number code. The BBC and Electron use COLOUR, followed by a number to indicate which colour.

PRINT (Commodore 64, Vic), PRINT TAB (BBC, Electron) and PRINT AT (Spectrum) are commands telling the computer to print something at a given location. The lines on the left contain a print command which indicates 5 lines down from the top of the screen and starting at the left-hand edge (0 is the first available space to print in).*

Sounds

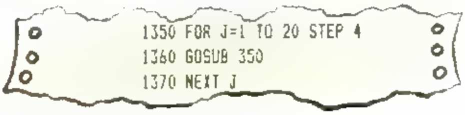
The other service subroutines are sounds. Sound commands vary, but they must give volume (how loud), pitch (how high) and duration (how long). This is easy to see on the BBC:


volume
pitch
duration

```
350 SOUND1,-15,J,2:RETURN
360 SOUND0,-15,7,3:RETURN
```

The Commodore 64 has to POKE the data for sounds into registers (special locations for sounds in the memory). Two sounds are used, a "beep" and a "swish". The Spectrum can only make "beep"-type sounds, so two different "beeps" give the different effects.

In the first line given above, a variable, J, has been used for the pitch, so you can give instructions like this:



```
1350 FOR J=1 TO 20 STEP 4
1360 GOSUB 350
1370 NEXT J
```

These lines will work on all the computers. This causes the pitch to rise from its starting point (1), to a much higher pitch (20), not by producing every sound on the scale, but only every fourth sound (STEP 4).

**Machine-specific methods of giving screen-handling commands are given on page 32.*

The conjure routine

The biggest subroutine in the game module is the conjure routine. In fact it consists of seven subroutines. This may seem a lot, but magic is an important part of a fantasy world and most character-types can use at least some spells to help them in their quest.

First the computer must check to see whether your character can use magic. Here is the subroutine which does this:

```
1000 PRINT tab(0,1);"YOU MAY USE MAGICKS";
1010 IF O(17)>0 THEN PRINT tab(0,2);"FROM NECRONOMICON";
1020 IF O(18)>0 THEN PRINT tab(0,3);"FROM THE SCROLLS";
1030 PRINT tab(0,4);"CONSULT THE LORE"
```

These lines check if you bought the Necronomicon, O(17), or Scrolls, O(18) and print a message.

```
1040 LET M$="USE SPELL NUMBER?":GOSUB 370
```

Sends computer to first type of message subroutine (see service subroutines, page 20).

```
1050 LET SL=VAL(I$)
```

SL = spell, VAL(I$) = number key pressed.

```
1060 IF SL=0 OR (O(17)=0 AND SL<5) OR
(O(18)=0 AND SL>3) OR SL>6 THEN GOTO 1040
```

If you press the number of a spell you have not got, computer asks you to try again.

```
1070 LET M(SL)=M(SL)-1:LET X=NX:LET Y=NY
```

Subtracts one "go" from your stock of the spell you have just used.

```
1080 IF M(SL)<0 THEN LET M$=T$(9):LET SL=7
```

Tells you if you have run out of a spell.

```
1090 FOR J=1 TO 5:PRINT tab(0,J);LEFT$(B$,W):NEXT J:GOSUB 570
1100 ON SL GOSUB 1140,1190,1220,1280,1300,1390,1130
```

Subroutines for each spell, as described below.

```
1110 LET F(5)=F(5)+.2
1120 GOSUB 430
```

Powersurge. Gives strength, F(1), and vitality, F(2), a boost beyond their starting values to make character extra strong.

```
1280 LET F(2)=F(2)+rnd(M(SL)):LET
F(1)=F(1)+rnd(M(SL)):LET F(7)=F(7)-1
1290 RETURN
```

Metamorphosis. Whatever is in this location becomes a different object at random.

```
1300 FOR J=1 TO 30
1310 LET R(NX,NY)=rnd(8)+1+C0
1320 GOSUB 350:GOSUB 570
1330 NEXT J
1340 IF RH<C7 THEN LET DX=255:LET MS=0
1350 FOR J=1 TO 20 STEP4
1360 GOSUB 350
1370 NEXT J
```

If it is used on a monster and new object is NOT another monster, this line tells the computer there is no longer a monster.

```
1390 LET F(2)=S2:LET F(1)=S1:LET
F(7)=F(7)-1
1400 RETURN
```

Healing. Restores strength and vitality to their starting values.

Sanctuary. If room character is in (RH) is empty, it turns into a protected space (C7) where he can hide.

```
1190 IF RH=C0 THEN LET R(NX,NY)=C7
1200 LET J=100:GOSUB 350:LET J=200:GOSUB 350
1210 RETURN
```

Teleport. Transports character to random new location. Old location becomes a blank.

```
1220 LET NX=rnd(13):LET NY=rnd(13)
1230 FOR J=0 TO 255 STEP8
1240 GOSUB 360:GOSUB 350
1250 NEXT J
1260 GOSUB 480
```

Here are the sound routines again.

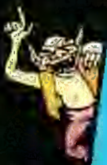
Superzap. These are sound subroutines. Computer loops through them twelve times.

```
1140 FOR J=1 TO 12
1150 GOSUB 350:GOSUB 360
1160 NEXT J
1170 IF DX<255 THEN LET X=MX:LET Y=MY:GOSUB 940
1180 RETURN
```

If there's a monster, computer goes to routine to "kill" it (remove it from screen).

More routines in the game module

Here you can see the remaining action subroutines in the game module. In these routines you will find that all the character's attributes and objects carried come into play.



Room contents

This routine is activated by the program itself, not by pressing a key. It reveals the contents of the room the character has just moved into. Remember the dungeon is dark until the character explores it.

```
580 LET RM=R(X,Y):PRINT tab(X,Y+5);CHR$(RM);
```

This line tells the computer to print a graphic symbol for the contents of location R(X,Y).

```
590 IF ABS(DX)<4 OR RM<=C7 THEN RETURN
```

These are conditions for activating a monster. If the object revealed is not a monster (less than or equal to C7) or there is already a monster within three squares of the character (DX=distance between the character and a monster), then the computer takes no action. If neither of these conditions applies, the next line activates the monster (see "Monster attacks" routine on page 23).

Reveal

A character who is carrying a light can reveal a section of the dungeon, if he chooses. An area of 3×3 squares will be lit up at once by activating the room-contents routine (above) for each location. If there is a monster in this area, it will be activated. Every time it is used, the computer subtracts one from the number of times the light can be used.

If the character has no light, this message tells him so.

```
1690 IF LT=0 THEN LET M$=T$(7):GOSUB430:
RETURN
1700 FOR Y=NY-3 TO NY+3
1710 FOR X=NX-3 TO NX+3
1720 IF (X)0 AND X<16) AND (Y>0 AND Y<16)
THEN GOSUB570
1730 NEXT X.NEXT Y
1740 LET LT=LT-1
1750 RETURN
```

Get

This routine allows the character to pick up objects in the dungeon.

```
1410 LET GX=NX+D(NF,1):LET GY=NY+D(NF,2)
```

This is the location the character can get something from. It is the square he's facing.

```
1420 IF GX<0 THEN LET GX=0
1430 IF GY<0 THEN LET GY=0
1440 IF GX>15 THEN LET GX=15
1450 IF GY>15 THEN LET GY=15
```

These lines ensure character can only get things within dungeon area on your screen.

```
1460 LET GT=R(GX,GY):IF GT>C1 AND GT<C4 THEN LET R(GX,GY)=C0
```

This line finds out what the character wants to get. Only C2 (potion), C3 (treasure) and C4 (Idol) are "gettable".

```
1470 IF GT=C2 THEN LET O(23)=O(23)+1:LET O(24)=O(24)+1
1480IF GT=C3 THEN LET TR=TR+1
1490 IF GT=C4 THEN GOSUB 1550
```

This line adds what the character gets to his stock of treasure or potions.

C4 is the Lost Idol. If it is found, then the quest is over.

```
1500 LET X=GX:LET Y=GY:GOSUB 570
1510 IF GT>C1 AND GT<C4 THEN LET J=GT:GOSUB 350:LET J=GT+5:GOSUB 350
1520 RETURN
```

This makes the object disappear from the screen, using the room-contents routine (see left).

This is a sound subroutine.

Potion

These lines revive your character's strength, F(1), and vitality, F(2), by using up his potions, O(23), and salves, O(24).

```
1660 IF O(24)>0 AND F(1)<S1 THEN LET F(1)=S1:LET O(24)=O(24)-1
1670 IF O(23)>0 AND F(2)<S2 THEN LET F(2)=S2:LET O(23)=O(23)-1
```

O=object array. F=array of attribute scores.

```
1680 RETURN
```

Monster dies

```
940 LET DX=255:LET MS=0:LET R(MX,MY)=C0
950 LET F(5)=F(5)+.1
960 LET M$=T$(6):GOSUB 430
970 FOR J=200 TO 150 STEP-8:GOSUB 350:GOSUB
360:NEXT J
980 GOSUB 570:RETURN
```

These lines de-activate the monster (saying that DX=255 means that the monster is no longer near enough to be activated) and make the space it occupied a blank. Character's experience, F(5), goes up and a message and sounds announce his success.

Monster attacks

```
600 LET MT=RM:LET M=MT-C2:LET MV=M/16:LET MS=M*6:LET
DX=3:LET LX=X:LET LY=Y
610 RETURN
```

If the room-contents routine (page 22) reveals a monster, then this line activates it. LX,LY = monster's location; MT = symbol of monster-type; M = monster-type for computer's calculations; MV = monster's moving speed; MS = monster's strength; DX = 3 means monster is near enough to character to be activated.

```
620LET DX=LX-MX:LET SX=SGN(DX):LET DY=LY-(MV*SY):LET
RM=R(MX,MY)
```

Finds distance between monster and character.

```
630 LET MX=LX-(MV*SX):LET MY=LY-(MV*SY):LET RM=R(MX,MY)
640 IF RM)C0 AND RM()MT THEN LET MY=LY:LET MX=LX
```

Moves monster, but prevents it crossing an occupied location.

```
650 LET R(LX.LY)=C0:LET X=LX:LET Y=LY:GOSUB 570
660 LET R(MX,MY)=MT:LET X=MX:LET Y=MY:GOSUB 570
670 LET LX=MX:LET LY=MY:LET H=0
```

Places a blank in old position of monster, puts monster symbol in new position and gives new monster starting-position.

```
680 IF ABS(DX)(=1 AND ABS(DY)(=1 AND RM()C7 THEN LET
H=M*.5:LET  J=H:GOSUB 350
```

If distance between monster and character is one square or less, monster hits UNLESS character is on a safe square, C(7); H = hit.

```
690 IF H*12(F(6)+F(3) THEN RETURN
700 LET M$=T$(5):GOSUB 430:GOSUB 360
```

Hit is successful if H ✱ 12 is more than luck score, F(6), added to agility score, F(3), of character. Message tells you hit is successful.

```
710 LET H=H/(3+O(9)+O(10)+O(11)+O(12)+O(13)+O(14))
720 LET F(1)=F(1)-H:LET F(2)=F(2)-(H/101)
```

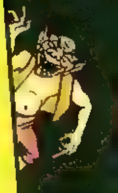
Hit value is divided by defensive value of these objects carried. Character's strength, F(1), and vitality, F(2), go down.

```
730 LET I=1:LET WB=0:LET MB=rnd(M)
740 LET J=MT:GOSUB 350:GOSUB 360
750 IF MB=1 AND O(1))0 THEN GOSUB 780
760 IF I(11 THEN LET I=I+1:GOTO 750
770 RETURN
780 LET O(I)=0:LET M$=T$(8)+" "+W$(I):GOSUB 430
790 LET MB=0:GOSUB 360:LET J=1:GOSUB 350
800 RETURN
```

If character is hit and still has a weapon, the subroutine at line 750 prints a message, "BROKEN THY" plus weapon name from W$.

```
810 LET NF=5:LET F(1)=0:GOSUB 440
820 PRINT tab(1,5);"THOU HAST EXPIRED!"
830 FOR J=150 TO 1 STEP-4
840 GOSUB 350:GOSUB 360:GOSUB 570:GOSUB 480
                                    850 NEXT J
                                    860 RETURN
```

Character is killed. Character symbol changes to ■ (NF = 5). His strength = 0. Sounds and messages tell you the character is dead.



Character attacks

This is what happens if your character attacks a monster. Here, objects and attributes not displayed on the screen influence the action.

```
870 LET M$=T$(rnd(3)):GOSUB 360
880 LET H=F(1)+O(1)+O(2)+O(3)+O(4)+O(5)+O(7)+O(8)
+rnd(F(6))
890 IF F(3)+F(6)(rnd(M)+2 THEN LET M$=T$(4):LET HT=0
900 LET MS=MS-H:GOSUB 430
910 LET F(1)=F(1)-(H/100):LET F(5)=F(5)+.05
920 IF MS(1 THEN GOSUB 940
```

LINE 870 prints message if monster is hit.

LINE 880 calculates strength of hit – it equals strength score, F(1), plus offensive value of any weapons carried.

LINE 890 If character's agility score, F(3), added to his luck score, F(6), is less than value of monster-type, then he misses.

LINE 900 subtracts value of hit from monster's strength score.

LINE 910 makes character's strength go down a bit and experience go up a bit as a consequence of the hit.

LINE 920 If monster's strength is below 1, it dies (see Monster dies routine).

Save game

These lines save the dungeon contents and character's attribute scores in the same way as in the dungeon generator and character creator, only the graphic symbol that represents the hero is saved, too.

```
2260 2260 LET M$="ONE MOMENT PLEASE":GOSUB 430
2270 LET S$="":LET T$=""
2280 FOR Y=1 TO 15
2290 FOR X=1 TO 15
2300 LET T$=T$+CHR$(R(X,Y))
2310 NEXT X
2320 NEXT Y
2330 LET T$=T$+CHR$(OS*NX)
2340 LET T$=T$+CHR$(OS+NY)
```

These lines save the dungeon by making string variable of the object-types in each location.

```
2350 LET S$=S$+CHR$(AS+OT)
2360 FOR I=1 TO 8
2370 LET S$=S$+CHR$(F(I)+AS)
2380 NEXT I
2390 FOR I=1 TO OT
2400 LET S$=S$+CHR$(O(I)+AS)
2410 NEXT I
2420 LET S$=S$+CHR$(GC+AS)
2430 LET S$=S$+CHR$(TR+AS)
```

And these lines save attribute scores and objects carried in another string variable.

```
2440 LET S$=S$+C$
2450 LET M$="ANY KEY TO SAVE":GOSUB 370
2460 OPEN 1,1,1,"HERO":PRINT£1,S$:CLOSE 1
2470 OPEN 1,1,1,"LEVEL":PRINT£1,T$:CLOSE 1
2480 LET FI=1                        2490 RETURN
```

De-bugging

It is not easy to type the *Dungeon of Doom* program listing into your computer. It is very long, and just one mistake, or bug, could prevent it working properly. So take it slowly and stop to check what you have typed every 20 lines or so. Take a break if you feel your concentration is slipping.

Common bugs to look out for

Whenever you are programming there are certain bugs that frequently crop up. Take care, for example, not to confuse zero with the letter "O", or the figure one with the letter "I". Remember that every letter, number, space and punctuation mark is important. If you leave any out, or add extra ones, the game will be altered and may not even work at all.

Long lines and brackets

In *Dungeon of Doom* there are a number of long program lines. The longer the line, the easier it is to miss something out so it is worth concentrating extra hard when you come across one. Calculating lines in the game module tend to be quite long and involved, like this one:

```
850 LET H=F(1)+O(1)+O(2)+O(3)
+O(4)+O(5)+O(7)+O(8)+rnd(F(6))
```

This line calculates the character's hit power. It uses a lot of brackets, which look confusing. Note especially the double bracket at the end. It is absolutely necessary to the sense of the line, so be sure not to miss it out here, or in other lines where it occurs.

Be sure to type exactly what is given.

Graphics behaviour problems

If objects behave in a strange way when you play, such as your character cannot pick up a "gettable" object, or can walk through walls, you should check to make sure that the object-type definitions C0-C8 (lines 2930-2940) refer to the right symbols, as the computer uses them to define the behaviour of objects.

Missing data

If you dimension an array and leave out one of the items of data that should go in it, your computer will look for the missing item in the next data line.

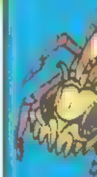
```
DIM C(8)
DATA 10,11,1,4,6,4,9,
FOR I=1 TO 8 READ C(I):NEXT I
DATA "SWORD","AXE","DAGGER".....
```

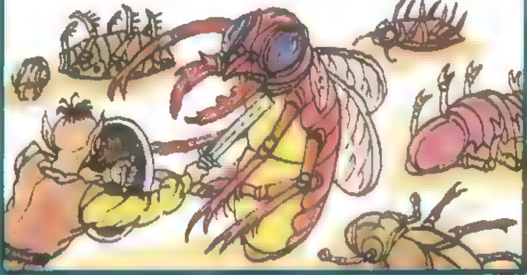
Number missing here.

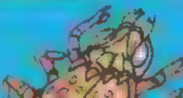
The computer will attempt to find the eighth item on this line. In this example you get a "TYPE MISMATCH" error as the computer tries to put a word into a number array. Even if the data is compatible, you will eventually get an error message as there will not be enough data to fill the arrays.

Mis-loading

If you load a dungeon when the game asks for a character (or the other way round) the computer will load what you feed it, but will not be able to make sense of it and you will get an unfamiliar arrangement of symbols on your screen. If this happens, you will have to reload the game module and start again.







Expanding the game

Once you are familiar with *Dungeon of Doom*, you will probably think of ideas for changing or improving it. Making alterations is a good way of practising programming. It's a good idea to make small changes first. If you make more complex changes remember to check every part of the program that is affected. This is especially important in *Dungeon of Doom* as there are three inter-relating parts.

Numbering new lines

You may need to expand arrays, set up new ones or add lines. If you add lines, be sure to number them to fit between existing lines :

```
190 IF NX>15 THEN LET NX=15
200 LET RH=R(NX,NY)
```

You can insert new lines labelled 191-199 in here if you want to.

Changing the Book of Lore

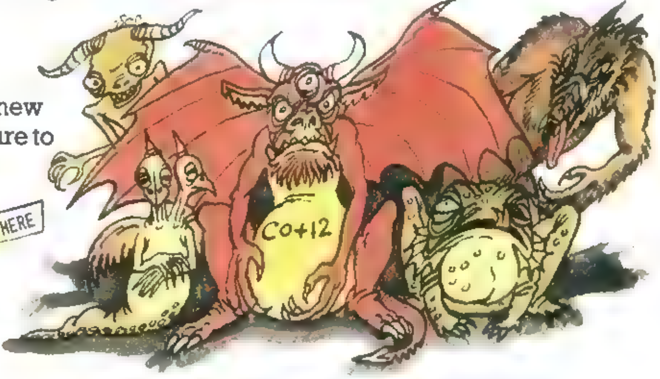
If any of the changes you make concerns which keys the player uses, you should make a note of them. The Book of Lore on pages 28-31 contains the rules. You should add any new ones on a sheet of paper, or make your own Book of Lore in an exercise book or notepad.

On the right of this page and on the next two pages you will find some possible program alterations that you can experiment with.

More monsters

Adding more monster-types will make dungeons more dangerous for your character. Here is what you do:

You must design a new monster symbol (see user-defined graphics on page 16) and add it to the other graphic symbols in the dungeon generator and game module. You must also change the lines that feed the data in (see how to do this on page 46). Your new monster will have a value of C0+12.



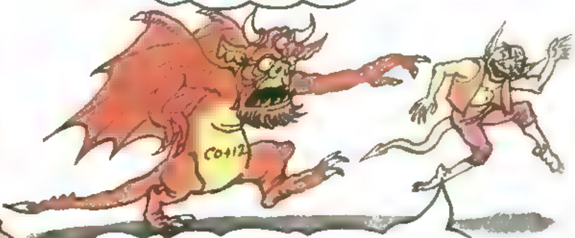
The game module already states that any object with a value greater than C0+8 will behave like a monster, so your new symbol will automatically do so.

You must also change line 240 in the dungeon generator to read:

```
240 IF I=9 THEN LET I=rnd(4)
```

This now allows for four monster-types.

See page 32 for the correct random command for your machine.



NOTE: A monster's speed and strength are decided by how much greater than C0 its code is (see line 650 in the game module), so the higher the code, the stronger and faster it is.

Additional room contents

To include extra types of room contents you must design and program new graphic symbols to represent them. This time you don't want them to act like monsters so you must change line 590 in the game module to prevent this:

```
590 IF ABS(DX)<4 OR (RM<=C7 OR RM>C8)
THEN RETURN
```

This new condition refers to symbols with a value higher than the last monster-type (C8).

Here's how you could add a torch symbol

1. Design your torch symbol, label it and add it to the dungeon generator and game module. It can be called C0+13.

2. You must allocate it a key in the dungeon generator, which you press to place it in a location. There are no spare number keys, so you will need to use a letter. Do it by adding a line:

```
155 IF I$="T" THEN R(X,Y)=C0+13
```

Now the "T" key will place torches in the dungeon.

3. The torch is "gettable" so you must allow for this in the Get routine in the game module. You should add a line like this after line 1490:

```
1495 IF GT=C0+13 THEN LET L=L+12
```

This adds to your stock of light.

4. You could try reducing the starting level of L (line 2240 in the game module) to make finding torches more important.

Secret squares

You can create secret traps or cause unexpected events by defining a new symbol as a blank, but with special conditions attached to it. It is "invisible" as it is the same colour as the background, but if a character tries to cross it he triggers some action as in the example below.

NOTE: Each type of secret square must be allocated a key in the dungeon generator, as explained for the torch symbol on this page. You can add the lines relating to each square after line 350 in the game module.

Rockfall

You can program a possible rockfall by adding a blank symbol, C0+14 and this line:

```
321 IF RH=C0+14 AND rnd(3)=1
THEN PRINT "ROCKFALL":R(X,Y)=C1
```

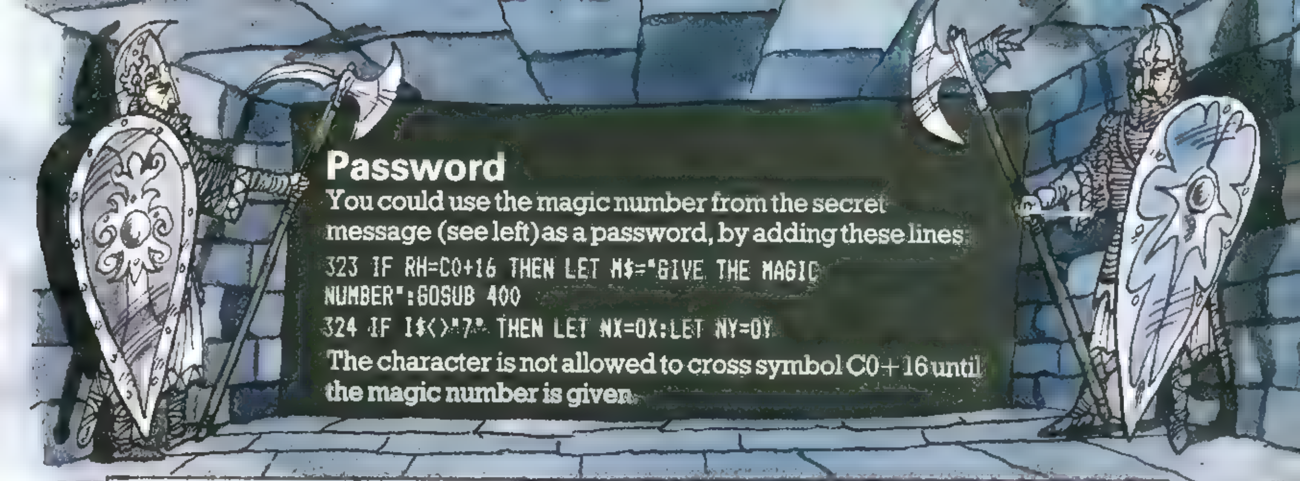
This provides a one in three chance that the way will be blocked if the character tries to enter a location containing this symbol. The room contents become a wall (C1). A message tells the character there has been a rockfall.

Secret message

Another idea is a blank square that activates a secret message. It could simply be a message of encouragement, or it could be something to be used elsewhere in the game, such as a password (see right).

```
322 IF RH=C0+15 THEN LET M$="THE MAGIC
NUMBER IS 7":R(X,Y)=C0:GOSUB 450
```

This line relates to a blank symbol C0+15. When crossed, it flashes a message, then becomes an ordinary blank.



Password

You could use the magic number from the secret message (see left) as a password, by adding these lines:

```
323 IF RH=C0+16 THEN LET M$="GIVE THE MAGIC
NUMBER":GOSUB 400
324 IF I$<>"7" THEN LET NX=OX:LET NY=OY
```

The character is not allowed to cross symbol C0+16 until the magic number is given.

Using attributes

You can make more out of the differences in attribute scores by using them as conditions for passing through certain locations.

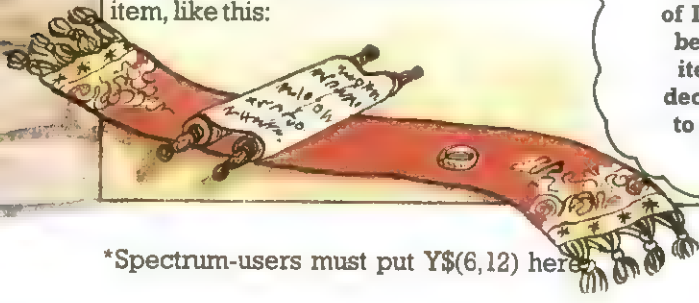
```
325 IF RH=C0+17 AND F(7)<5 THEN LET NX=OX:
LET NY=OY
326 LET M$="MAGIC BARRIER":GOSUB 440
```

NO ENTRY TO CHARACTERS WITH A MORALITY SCORE OF LESS THAN 5.

These lines stop the character crossing a location containing a symbol C0+17 if his morality score, F(7), is less than five. A message tells the player why. This forces different character-types to take different routes through a dungeon. Don't forget to allow alternatives, though.

Using magic items

At the moment, all six spells relate to either the Necronomicon or Scrolls when you play the game, although there are six magic items for sale in the trading post. You could make one spell relate to each item, like this:



Replace lines 1010 and 1020 in the game module with these:

```
1010 FOR J=17 TO 22:IF O(J)>0 THEN LET
M$=Y$(J-16);:GOSUB 450
1020 NEXT J
```

These lines now check which magic items the character is carrying (objects 17-22) and looks in Y$ for their names, which it then prints after the first line of the message, "YOU MAY USE MAGICKS FROM".

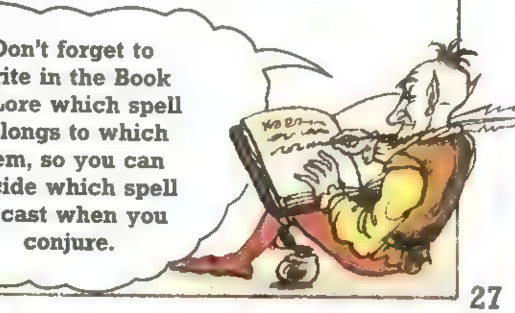
Y$
Don't forget to add Y$ to initialization in the game module.

Dimension Y$ by adding Y$(6)* to line 2520, then list the data, like this:

```
2555 DATA "NECRONOMICON","SCROLLS","RING",
"AMULET","SASH","CLOAK"
2556 FOR I=1 TO 6:READ Y$:NEXT I
```

The line in the game module that checks if you have a particular spell number must also be changed:

```
1060 IF SL=0 OR (O(17)=0 AND SL<5) OR (O(18)
=0 AND SL>3) OR SL>6 THEN GOTO 1040
```


Don't forget to write in the Book of Lore which spell belongs to which item, so you can decide which spell to cast when you conjure.

---

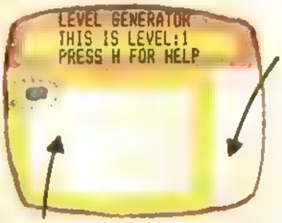
*Spectrum-users must put Y$(6,12) here.

BOOK OF LORE

These pages contain all you need to find your way in "Dungeon of Doom". Don't worry if there seems a lot to learn. You use the three parts of the program separately so you can familiarize yourself with them one at a time. Allow yourself some practice games to get used to playing.

The dungeon generator

When you want to make dungeons, load the dungeon generator program from its cassette. When the computer is ready it will set up the screen like this:



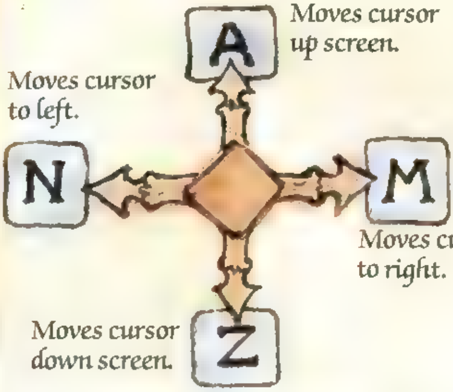
```
LEVEL GENERATOR
THIS IS LEVEL:1
PRESS H FOR HELP
```

Cursor can't move into this area. It is outside the dungeon plan.

This is the dungeon plan area. A flashing cursor appears at the top left.

Help

If you press H for help, a message tells you to "PRESS ANY KEY". Do so, and you will be reminded which keys to use to plan your dungeon. The information appears one line at a time and you must press any key again to read the next line. Here is the information you will be given:



Moves cursor up screen.

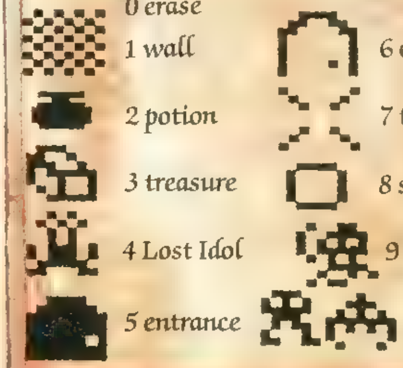
Moves cursor to left.

Moves cursor to right.

Moves cursor down screen.

Move the cursor into position using these keys. Then press a number key to place a symbol on the screen. The symbols and their number keys are shown at the top of this page, on the right.

Here are the symbols and their number keys:



- 0 erase
- 1 wall
- 2 potion
- 3 treasure
- 4 Lost Idol
- 5 entrance
- 6 exit
- 7 trap
- 8 safe place
- 9 monsters

Warning

The computer memorizes the location of the entrance symbol and places the character there when he enters the dungeon. If you place more than one entrance, the character will start on the location of the last one.

The Idol should only appear in the last of a series of dungeons, as the object of the Quest.

Save dungeon and escape the program

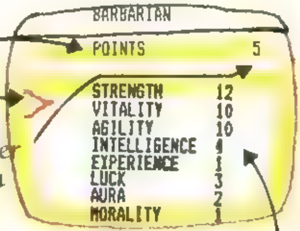
When you are happy with your dungeon, turn your cassette over and save it on the other side from the dungeon generator program. Press S to show you are ready. You will be asked to wait "ONE MOMENT PLEASE", then be told, "ANY KEY TO SAVE" and when you press a key your dungeon will be saved and the computer will prepare for you to design the next level. You will not be able to save the dungeon if it hasn't got an entrance and at least one exit so that the character can enter and leave the level. To escape the program press F.

Dungeon levels

You should design a series of dungeons of increasing difficulty, with the Idol in the last one. Each series should be planned at one "sitting" as they are numbered consecutively by the computer, but it will start at 1 again if you turn the computer off. Three to five levels should be enough for a good game.

The character creator

When you want to make a character, load the character creator program from its cassette. When it is ready it will set up the screen like this:



```
BARBARIAN
POINTS            5

STRENGTH     12
VITALITY     10
AGILITY      10
INTELLIGENCE  4
EXPERIENCE    1
LUCK          3
AURA          2
MORALITY      1
```

Message box

Cursor

Random number of modification points.

Attribute scores given by the computer.

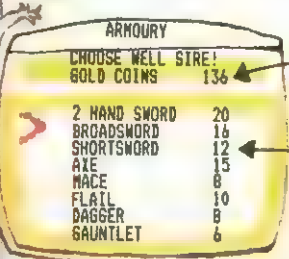
A moves the cursor up the screen.

Z moves the cursor down the screen.

Move the cursor until it indicates the attribute you want to change.
"+" adds to the score (points are deducted from available modification points).
"−" subtracts from the score (these points then become available as modification points).

As soon as you press a key the name of the character-type appears in the message box. If you make big changes to one or two of the scores, the character-type will probably change.

When you are happy with the attribute scores, press the spacebar to turn to the next page. The following three pages look like this:



```
ARMOURY
CHOOSE WELL SIRE!
GOLD COINS      136

2 HAND SWORD    20
BROADSWORD      16
SHORTSWORD      12
AXE             15
MACE             8
FLAIL           10
DAGGER           8
GAUNTLET         6
```

Gold coins

Prices

This screen shows the Armoury, where weapons and armour are for sale. The other two pages offer Accoutrements (necessities for the trip) and magic from the Emporium. The number of gold coins you have left appears at the top of each new page.

A and Z move the cursor again.
Press "+" to buy an item at full price.
Press "−" if you want to bargain. This is what happens then:

Computer	YOUR OFFER?
Player	Type a number and RETURN
Computer	IT'S YOURS or OFFER REJECTED
Player	If successful, go to next item. If not, press "−" to try again or "+" to pay the full price.

Messages and save character

If you indicate an item your character-type is not allowed to buy, you'll be told something like,
"NOT FOR MAGE".
If you try to buy more than one of any item other than potions and salves, a message appears,
"YOU HAVE IT SIRE".
When you have finished, press spacebar and you will be asked to,
"NAME THY CHARACTER".

Before you do so, turn your cassette over to where you want to save the character. Then invent a name with up to ten letters and press RETURN and your character will be saved.

"Dungeon of Doom" character-types

This chart gives you a rough guide to the character-types in "Dungeon of Doom". They can use strength, magic, or a combination of both in their Quest, and each type has advantages and disadvantages, as you will discover if you compare their performances in the same dungeon.

Character	Strength	Aura (magic)
Barbarian	v. high	v. low (none)
Warrior	high	low (none)
Wanderer	medium	medium (some)
Cleric	low	high (some)
Mage	v. low	v. high (all)

The game module

When you are ready to start your Quest, load the game module from its cassette. When the computer is ready you will get the message, "PREPARE HERO TAPE". Place the character creator cassette in your recorder and find the character you want to load.

Your next instruction is, "PRESS ANY KEY". When you do so, the character will be loaded. It will take 10-15 seconds depending on your equipment.

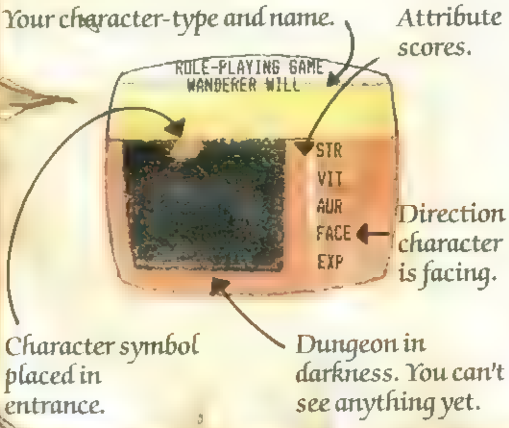
> You can't play until you have designed at least one character and one dungeon.

Next, you "PREPARE DUNGEON TAPE" and "PRESS ANY KEY" to load a dungeon. If the dungeon level you load is too difficult for the character's experience, you will be told, "LEVEL TOO DEEP. REWIND TAPE TO POSITION FOR LEVEL 2". (The level given will match the character's experience score.)

If you leave the dungeon cassette in your recorder, it should be in the right place to load the next dungeon when you need it.

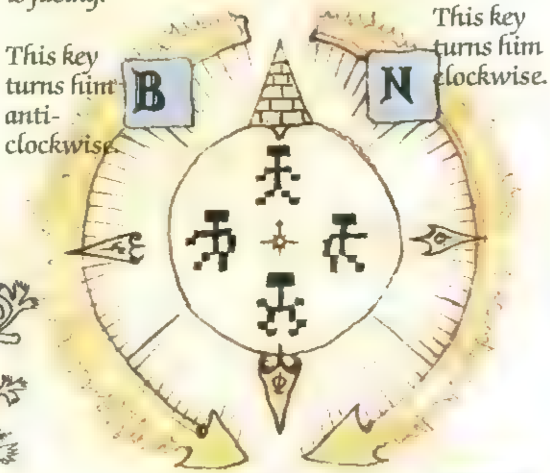
> Don't forget to remove the dungeon cassette if you want to save a game (see page 31).

When the preparations are complete, the screen will be set up like this:


Your character-type and name.
Attribute scores.
ROLE-PLAYING GAME
WANDERER WILL
STR
VIT
AUR
FACE
EXP
Direction character is facing.
Character symbol placed in entrance.
Dungeon in darkness. You can't see anything yet.

Keys you use to play

A character can only move in the direction he is facing.


This key turns him anti-clockwise.
B
N
This key turns him clockwise.

Change direction if you want to, then press M to move. M

Reveal

As the character explores the dungeon, the space he enters lights up and remains lit. He can enter any location except those containing a wall or a monster. Walls block the way and monsters will attack.

If you press R, you will reveal an area of 3x3 squares. BE WARNED: a monster revealed by the light is activated and will give chase and attack. It cannot cross occupied spaces, though, so your character can "hide" behind objects in the dungeon. If there are two monsters, only the nearest one gives chase.

Attack

To attack, press A and keep pressing while the fight goes on. Messages at the top of the screen report your progress. REMEMBER: watch the attribute scores. If your character's strength is running out, you may use magic (see page 31) or take a potion (see below) to recover.

Potions and salves

Press P to take a potion or salve at any time. One of each is used up whenever you press this key. They restore strength and vitality.

Conjure

If your character is a magic-user and you press C, the action is frozen and a message appears,

"YOU MAY USE
MAGICKS FROM
NECRONOMICON
FROM THE SCROLLS
CONSULT THE LORE
USE SPELL NO?"

Either, or both of these lines may appear, depending on which magic items your character has.

Consult this page and choose a spell from those on the right.

Get

Press G to get or pick up objects. Your character must be next to and facing the thing he wants to get. When he gets the Lost Idol, the Quest is over.

Save game

S saves the state of play. The other side of the game module cassette is a good place to save it. When you re-load, you start again where you left off.

Traps

Traps are "sticky" and hold a character back. If he crosses one, his strength is sapped and he may not be able to move for a moment. In this case, keep pressing "M" to move and he should break free. If he is very weak, taking a potion should help.

Attribute scores

Attribute scores change all the time. Even moving through the dungeon causes the strength score to flicker. Standing still restores strength slowly.

All exploits in the dungeon add a fraction to the experience score. When these add up to a whole number the score goes up and the character can exit to the next level.

To leave, place the character symbol on the exit symbol. You will be asked to "PREPARE DUNGEON LEVEL" and you should load the next dungeon in your series. Trying to leave before the experience score has increased will result in a message, "YOU NEED EXPERIENCE".

Magic spells

The possessor of the Necronomicon can use spells I to III. They are very powerful. Here is what they do:

I Superzap (key 1)

Monster is slain instantly by a magic blow.

II Sanctuary (key 2)

Creates a safe place in the location where the character is standing. Here, no monster can touch him and he can regain his strength until he is ready to make his escape.

III Teleport (key 3)

Magically whisks the character from his present location to a different one, at random. DANGER: this spell is unpredictable. If the user tries to materialize in a wall it will not work and he will stay where he was. Or, it may not transfer him far enough to be out of danger.

Spells IV to VI are contained in the Scrolls. They are less dramatic, but are useful in a tight spot.

IV Powersurge (key 4)

Gives a huge boost to the user's strength and vitality so he can fight more effectively.

V Metamorphosis (key 5)

Transforms whatever is in the user's location into some other object. WARNING: the effect is random. It can be used against monsters but there is a chance that another monster-type will appear in its place.

VI Healing (key 6)

A mild spell that restores strength and vitality to their original values. It is especially useful if the character has used up all his potions and salves.

Dungeon of Doom program listings

On the following pages you will find the listings for the three separate programs that make up the *Dungeon of Doom*. They have been written to run on the Commodore 64, Vic, BBC, Electron and Spectrum, although some conversion lines are necessary for each computer. The conversions are given at the end of each part of the program and lines that need to be converted are marked with symbols, like this:

Symbols key

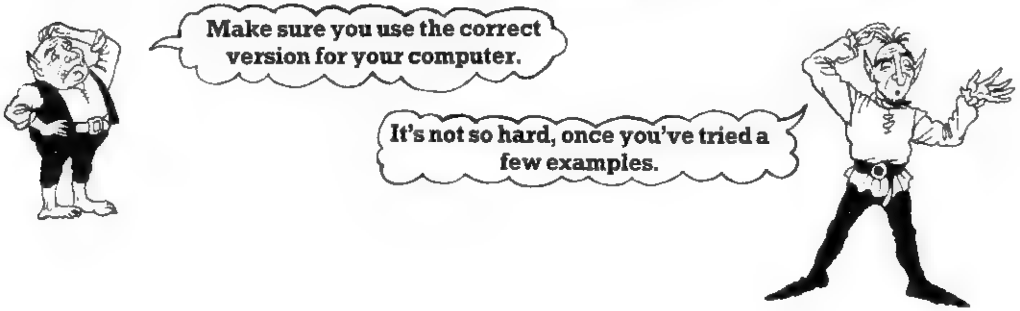
BBC/Electron ★ Vic ▲
C64 ■ Spectrum ○

In addition, any part of a line written in lower case (PRINT tab, for example) must be replaced with a machine-specific command for your computer. The general conversion chart below shows you how to do this, with examples to make it clear.

☆ The general conversions shown below are marked with this symbol in the listings.

		EXAMPLE
COMMAND	tab(X,Y);	PRINT tab(NX,NY+5);M$
BBC/ELECTRON	TAB(X,Y);	PRINT TAB(NX,NY+5);M$;
SPECTRUM	AT Y;X	PRINT AT NY+5,NX;M$
C64/VIC	HM$;LEFT$(CU$,Y); SPC(X);	PRINT HM$;LEFT$(CU$,NY+5) ;SPC(NX);M$ SPC((X):
COMMAND	X$=inkey$	LET I$=inkey$
BBC/ELECTRON	X$=INKEY$(0)	LET I$=INKEY$(0)
SPECTRUM	X$=INKEY$	LET I$=INKEY$
C64/VIC	GET X$	GET I$
COMMAND	paper X	paper BG
BBC/ELECTRON	COLOUR X+128	COLOUR BG+128
SPECTRUM	PAPER C(X+1)	PAPER C(BG+1)
C64/VIC	PRINT BG$(X);	PRINT BG$(BG);
COMMAND	ink X	ink 2
BBC/ELECTRON	COLOUR X	COLOUR 2
SPECTRUM	INK C(X+1)	INK C(3)
C64/VIC		SEE SPECIAL NOTE BELOW.
COMMAND	rnd(X)	G=rnd(F)+10
BBC/ELECTRON	RND(X)	G=RND(F)+10
SPECTRUM	INT(RND*X+1)	G=INT(RND*F+1)+10
C64/VIC	INT(RND(1)*X+1)	G=INT(RND(1)*F+1)+10

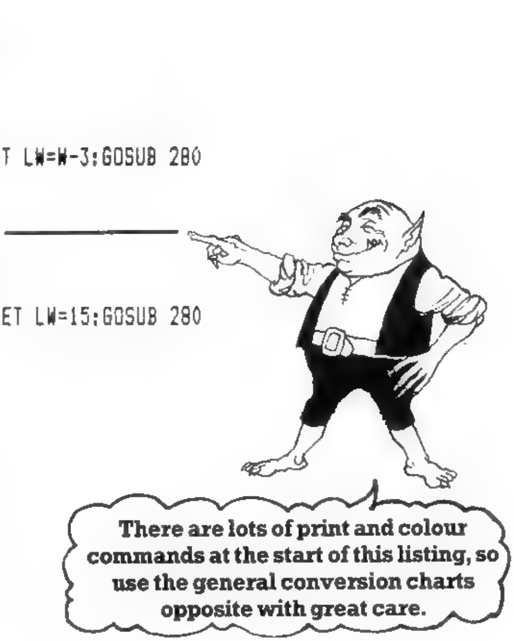
SPECIAL NOTE FOR C64/VIC USERS
Leave out the ink command UNLESS it is preceded by paper 0. In this case, use:PRINT BG$(X);PRINT BG$(0);
EXAMPLE: paper 0;ink 3 becomes PRINT BG$(3);:PRINT BG$(0);


Make sure you use the correct version for your computer.
It's not so hard, once you've tried a few examples.

The Dungeon Generator

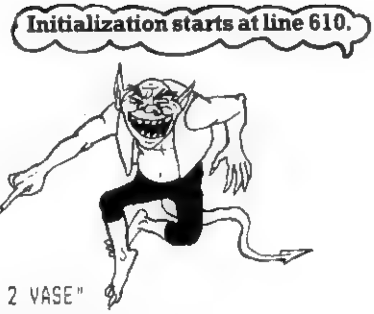
▲ See special note for Vic users page 35.

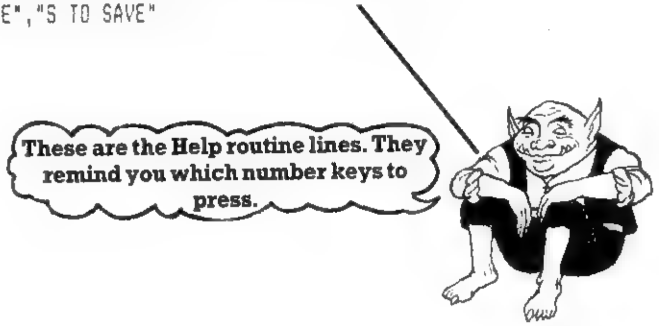
```
■    10 GOSUB 610
■▲☆  20 paper 3:CLS
     30 LET BG=2:LET FG=1:LET T=0:LET L=3:LET LW=W-3:GOSUB 280
☆    40 paper 2:ink 0
☆    50 PRINT tab(1,1);"LEVEL GENERATOR";
☆    60 PRINT tab(1,2);"THIS IS LEVEL:";LE;
☆    70 PRINT tab(1,3);"PRESS H FOR HELP"
     80 LET BG=3:LET FG=2:LET T=5:LET L=15:LET LW=15:GOSUB 280
     90 LET X=1:LET Y=1
☆   100 LET I$=inkey$
    110 IF I$="H" THEN GOSUB 360
    120 IF I$="A" AND Y>1 THEN LET Y=Y-1
    130 IF I$="Z" AND Y<15 THEN LET Y=Y+1
    140 IF I$="N" AND X>1 THEN LET X=X-1
    150 IF I$="M" AND X<15 THEN LET X=X+1
    160 IF I$>"/" AND I$<":" THEN GOSUB 230
☆   170 paper 3:ink 0
☆   180 PRINT tab(X,Y+5);CHR$(OS);
☆   190 PRINT tab(X,Y+5);CHR$(R(X,Y));
    200 IF I$="S" AND IX>0 THEN GOSUB 450:GOTO 20
    210 IF I$<>"F" THEN GOTO 100
    220 STOP
    230 LET I=VAL(I$)
☆   240 IF I=9 THEN LET I=8+rnd(3)
    250 IF I=5 THEN LET IX=X:LET IY=Y
    260 LET R(X,Y)=CO+I
    270 RETURN
☆   280 PRINT tab(0,T);
○☆  290 paper FG:PRINT LEFT$(B$,LW+2)
☆   300 paper BG:ink FG
    310 FOR I=1 TO L
○■▲ 320 PRINT TAB(0);CHR$(OS);LEFT$(B$,LW);CHR$(OS)
    330 NEXT I
○☆  340 paper FG:PRINT LEFT$(B$,LW+2);
    350 RETURN
☆   360 paper 1:ink 3
    370 FOR H=1 TO 10
☆   380 PRINT tab(1,4);H$(H);:GOSUB 430
○☆  390 PRINT tab(1,4);LEFT$(B$,W-2);
    400 NEXT H
■▲☆ 410 ink 3
    420 RETURN
☆   430 LET G$=inkey$:IF G$="" THEN GOTO 430
    440 RETURN
☆   450 PRINT tab(1,4);"ONE MOMENT PLEASE.";
    460 LET S$=""
    470 FOR J=1 TO 15
    480 FOR K=1 TO 15
    490 LET S$=S$+CHR$(R(K,J))
```


There are lots of print and colour commands at the start of this listing, so use the general conversion charts opposite with great care.

It may be helpful to use a ruler to mark your place as you type lines in, so that you can find where you were easily after looking away from the page.

```
    500 NEXT K
    510 NEXT J
    520 LET S$=S$+CHR$(IX+OS):LET S$=S$+CHR$(IY+OS)
    530 LET S$=S$+CHR$(LE+OS)
 ☆  540 PRINT tab(1,4);"ANY KEY TO SAVE  ";:GOSUB 430
○■▲ 550 LET S=OPENOUT "LEVEL"
 ■▲
○■▲ 560 PRINT#S,S$
 ■▲
○■▲ 570 CLOSE#S
 ○☆ 580 PRINT tab(1,4);LEFT$(B$,W)
    590 LET LE=LE+1:GOSUB 700
    600 RETURN
  ○ 610 DIM R(15,15),H$(10)
  ○ 620 GOSUB 790
    630 DATA "PRESS ANY KEY","TO MOVE A Z N M","1 WALL   2 VASE"
    640 DATA "3 CHEST 4 * IDOL *","5 WAY IN  6 EXIT","7 TRAP","8 SAFE PLACE"
    650 DATA "9 GUARD","0 TO ERASE","S TO SAVE"
    660 LET LE=1
    670 FOR I=1 TO 10
    680 READ H$(I)
    690 NEXT I:GOSUB 810
    700 FOR J=1 TO 15
    710 FOR K=1 TO 15
    720 LET R(J,K)=CO
    730 NEXT K
    740 NEXT J
    750 LET IX=0:LET IY=0
    760 LET B$="":FOR I=1 TO W:LET B$=B$+" ":NEXT I
    770 RETURN
■○▲★ 790 REM SET UP THE CHAR START
   ★ 800 RETURN
    810 REM READ THE CHARACTERS
```


Initialization starts at line 610.


These are the Help routine lines. They remind you which number keys to press.

▲○■★ Add lines 820-840/860 as shown in the conversions for your computer.

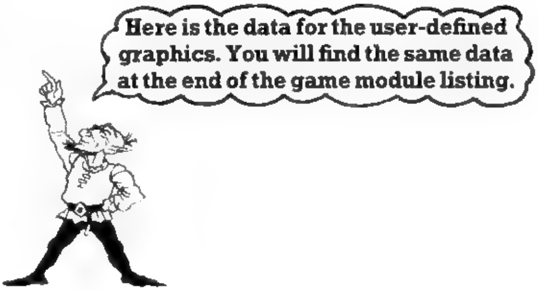
```
1000 DATA 255,255,255,255,255,255,255,255
1010 DATA 0,0,0,0,0,0,0,0
1020 DATA 85,170,85,170,85,170,85,170
1030 DATA 0,60,24,60,126,126,126,60
1040 DATA 0,56,100,114,95,73,41,31
1050 DATA 20,42,20,20,93,93,62,99
1060 DATA 60,126,255,255,255,253,255,255
1070 DATA 60,102,195,129,129,129,133,129
1080 DATA 129,66,36,0,0,36,66,129
1090 DATA 0,60,66,66,66,66,60,0
1100 DATA 76,158,170,190,84,30,37,88
1110 DATA 0,56,84,124,56,44,68,102
1120 DATA 0,8,28,42,127,85,65,34
```

▲■ Add lines 4000-4030 from page 35.

■ Add lines 5000-5080 from page 35.

BBC/Electron ★

```
790 OS=224:CO=OS+6:W=20
795 MODE 5:VDU 23,0,8202;0;0;0;
820 VDU 23,224:FOR I=0 TO 7:
    READ A:VDU A:NEXT I
830 FOR I=0 TO 11:VDU 23,230+I
840 FOR J=0 TO 7:READ A:VDU A:NEXT J
850 NEXT I:RETURN
```


Here is the data for the user-defined graphics. You will find the same data at the end of the game module listing.

Conversions

Below you will find the conversions you need for your computer. Some of the lines need to be included in two or even all three parts of the program, so read the instructions carefully. When this happens, you will be referred back to this page to add the required lines.

Commodore 64/Vic ■▲

SPECIAL NOTE FOR VIC USERS ▲

```
Before typing in or loading this program type in the following:
POKE 44,28:POKE 642,28:SYS(64824)
```

■▲

```
320 PRINT BG$(FG);" ";BG$(BG);LEFT$
    (B$,LW);BG$(FG);" "
410 Leave out this line
550 OPEN 1,1,1,"LEVEL"
555 FOR I=0 TO 2
560 PRINT#1,MID$(S$,I*76+1,76)
565 NEXT I
570 CLOSE 1
```

These lines must also be included on the C64 ONLY. ■

```
  5 GOSUB 5000
 20 PRINT CHR$(147):POKE 53280,0:POKE 53281,0
790 OS=96:CO=OS+6:W=40:GOSUB 4000
820 FOR I=0 TO 7:READ A:POKE 36352+I,255-A:NEXT I
830 FOR I=0 TO 95:READ A:POKE 36400+I,255-A:NEXT I
840 RETURN
```

The following lines must be used on the Vic ONLY. ▲

```
 20 PRINT CHR$(147):POKE 36879,8
790 OS=96:CO=OS+6:W=22:GOSUB 4000
820 FOR I=0 TO 2047:POKE 5120+I,PEEK(32768+I):NEXT I
830 FOR I=0 TO 7:READ A:POKE 6656+I,255-A:NEXT I
840 FOR I=0 TO 95:READ A:POKE 6704+I,255-A:NEXT I
850 POKE 36869,205
860 RETURN
```

Lines 4000-4030 must be added to ALL THREE PROGRAMS for the C64 and Vic. ■▲

```
4000 BG$(0)=CHR$(146):BG$(1)=CHR$(18)+CHR$(28)
4010 BG$(2)=CHR$(18)+CHR$(158):BG$(3)=CHR$(18)+CHR$(5)
4020 HM$=CHR$(19):CU$="":FOR I=1 TO W:LET CU$=CU$+CHR$(17):NEXT I
4030 POKE 650,255:RETURN
```

Lines 5000-5080 must be added to the dungeon generator and the game module for the C64. ■

```
5000 POKE 52,128:POKE 56,128
5010 POKE 56334,PEEK(56334) AND 254:
     POKE 1,PEEK(1) AND 251
5020 FOR I=0 TO 2047:POKE 34816+I,
     PEEK(53248+I):NEXT I
5030 POKE 1,PEEK(1) OR 4
5040 POKE 56334,PEEK(56334) OR 1
5050 POKE 56578,PEEK(56578) OR 3
5060 POKE 56576,(PEEK (56576)
     AND 252)OR 1
5070 POKE 53272,2:POKE 648,128
5080 RETURN
```

Spectrum○

```
290,320,340,390,580 Replace LEFT$(B$,Exp) with
    B$(TO Exp)e.g.in line 390 use B$(TO W-2)
550 Leave out this line
560 LET Q$(1)=S$
570 SAVE "LEVEL" DATA Q$()
610 DIM R(15,15):DIM H$(10,18)
615 DIM Q$(1,228)
790 LET W=32:LET OS=144:LET CO=OS+6
820 FOR I=0 TO 7:READ A:POKE USR"A"+I,A:NEXT I
830 FOR I=0 TO 95:READ A:POKE USR "G"+I,A:NEXT I
840 DIM C(4)
850 LET C(1)=0:LET C(2)=2:LET C(3)=6:LET C(4)=7
860 RETURN
```

Character Creator

Here is the listing for the character creator. Conversion lines are marked (see key on page 32) and you should look up the lines you need for your computer on page 39.

```
       10 GOSUB 1060
▲○■☆ 20 paper 0:CLS
   ■▲ 30 LET J=1:LET H=MP:LET H$="POINTS"
       40 GOSUB 810:GOSUB 900
       50 LET K=1:LET P=T+1
     ☆ 60 PRINT tab(1,P);">";
       70 GOSUB 720
       80 IF K=5 THEN GOTO 70
 ○▲■ 90 IF I$=";" AND H>0 THEN LET F(J,K)=F(J,K)+1:LET H=H-1:GOSUB 920
     ○100 IF I$="-" AND F(J,K)>1 THEN LET F(J,K)=F(J,K)-1:LET H=H+1:GOSUB 920
      110 LET C=1
      120 IF F(1,4)>6 AND F(1,8)>7 THEN LET C=2
      130 IF F(1,4)>8 AND F(1,7)>7 THEN LET C=3
      140 IF F(1,1)>7 AND F(1,8)>5 AND F(1,1)+F(1,2)>10 THEN LET C=4
      150 IF F(1,1)>8 AND F(1,2)+F(1,3)>12 AND F(1,8)<6 THEN LET C=5
      160 LET M$=C$(C)
      170 GOSUB 860
      180 IF I$<>" " THEN GOTO 70
      190 LET H=GC:LET H$="GOLD COINS:"
      200 FOR J=2 TO 4
      210 LET K=1:LET P=T+1
      220 LET M$="CHOOSE WELL SIRE!"
      230 GOSUB 810
      240 GOSUB 900
     ☆250 PRINT tab(1,P);">";
      260 GOSUB 720
      270 LET N=8*(J-2)+K
      280 LET M$="MAKE YOUR CHOICE"
      290 GOSUB 680
      300 LET BR=0:LET OF=0
   ○▲■310 IF I$=";" THEN LET OF=F(J,K):GOSUB 610
    ○☆320 IF I$="-" THEN LET BR=rnd(3):GOSUB 570
      330 GOSUB 860
      340 IF I$<>" " THEN GOTO 260
      350 NEXT J
     ☆360 PRINT tab(1,2);"NAME THY CHARACTER";
    ○☆370 PRINT tab(1,3);LEFT$(B$,W-2);:PRINT tab(1,3);
    ▲■380 INPUT N$
      390 IF LEN(N$)>10 THEN GOTO 360
     ☆400 PRINT tab(1,3);"ONE MOMENT PLEASE";
     ☆410 PRINT tab(1,3);
      420 LET D=D*3
      430 LET S$=CHR$(D+AS)
      440 FOR I=1 TO 8
      450 LET S$=S$+CHR$(F(1,I)+AS)
      460 NEXT I
```

Look up the PRINT command for your computer on page 32 every time you see a command like this.

This line gets the computer to create a random bargain price as much as three gold coins less than the original price. The random command varies. Find the right one for your computer from this box.

This line equals 0 at the moment as it will be used for counting treasure and your character does not have any when he starts the game. The line must be included, though, or the computer won't know where to store treasure when it is found.

```
   470 FOR I=1 TO 0
   480 LET S$=S$+CHR$(O(I)+AS)
   490 NEXT I
   500 LET S$=S$+CHR$(H+AS)
   510 LET S$=S$+CHR$(AS)
   520 LET S$=S$+N$+" -"+C$(C)
○▲■530 LET S=OPENOUT "HERO"
○▲■540 PRINT#S,S$
○▲■550 CLOSE#S
   560 STOP
   570 LET M$="":GOSUB 860
  ☆580 PRINT tab(2,2);"YOUR OFFER";
 ▲■590 INPUT OF
   600 GOSUB 680
   610 IF O(N)>0 AND N<23 THEN LET M$="YOU HAVE IT SIRE":RETURN
   620 LET PR=F(J,K)-BR
   630 IF H<PR THEN LET M$="YOU CANNOT AFFORD":RETURN
   640 IF OF>=PR AND Y=1 THEN LET O(N)=O(N)+P(N):LET H=H-PR:LET M$="TIS YOURS'"
   650 IF OF<PR AND Y=1 THEN LET M$="OFFER REJECTED"
   660 IF H<0 THEN LET H=0
   670 RETURN
   680 LET Y=0
  ○690 IF MID$(O$(N),C,1)="1" THEN LET Y=1
   700 IF Y=0 THEN LET M$="NOT FOR "+C$(C)
   710 RETURN
  ☆720 LET I$=inkey$
   730 IF I$="" THEN GOTO 720
  ☆740 paper 3:ink 1
  ☆750 PRINT tab(1,P);" ";
   760 IF I$="A" AND K>1 THEN LET K=K-1
   770 IF I$="Z" AND K<D THEN LET K=K+1
   780 LET P=K*2+T-1
  ☆790 PRINT tab(1,P);">";
   800 RETURN
  ☆810 paper 0:ink 2
 ○☆820 PRINT tab(0,0);LEFT$(B$,W);
  ☆830 PRINT tab(0,0);F$(J,9)
   840 LET BG=2:LET FG=3:LET T=1:LET L=2
   850 GOSUB 980
  ☆860 paper 2:ink 0
 ○☆870 PRINT tab(2,2);LEFT$(B$,17);tab(2,2);M$;
 ▲■
  ☆880 PRINT tab(2,3);H$;tab(15,3);H;" ";
   890 RETURN
   900 LET BG=3:LET FG=2:LET T=5:LET L=15
   910 GOSUB 980
  ☆920 paper 3:ink 0
   930 FOR I=1 TO 8
   940 LET Y=T+(I-1)*2+1
 ▲■
  ☆950 PRINT tab(2,Y);F$(J,I);tab(16,Y);F(J,I);" ";
```

For machine-specific colour commands see chart on page 32.

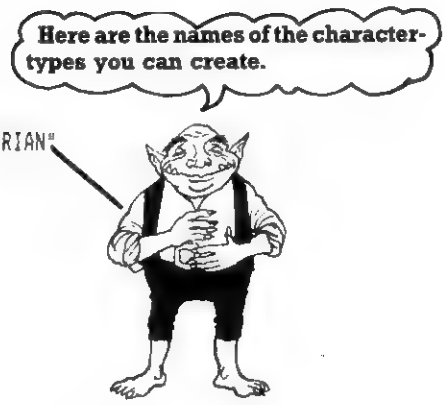
```
    960 NEXT I
    970 RETURN
  ☆ 980 PRINT tab(0,T);
 ○☆ 990 paper FG:PRINT LEFT$(B$,W);
  ☆1000 paper BG:ink FG
   1010 FOR I=1 TO L
○▲■1020 PRINT CHR$(B);LEFT$(B$,W-2);CHR$(B);
   1030 NEXT I
 ○☆1040 paper FG:PRINT LEFT$(B$,W);
   1050 RETURN
   1060 GOSUB 1600
   1070 LET D=8
   1080 DIM F(4,D+1)
  ○1090 DIM F$(4,D+1)
  ○1100 DIM C$(5)
   1110 DIM O(D*3)
  ○1120 DIM O$(D*3)
   1130 DIM P(D*3)
   1140 DATA "00001","00011","10011","10011","10011","00011","11111","10011"
   1150 DATA "00011","00011","10011","11111","00011","11011","11011","11111"
   1160 DATA "11100","00100","11100","10100","11100","11100","11111","11111"
   1170 FOR I=1 TO D*3
   1180 READ O$(I)
   1190 NEXT I
   1200 FOR I=1 TO 8
  ☆1210 LET F(1,I)=rnd(5)+2
   1220 NEXT I
   1230 LET F(1,5)=1
   1240 DATA 20,16,12,15,8,10,8,6
   1250 DATA 18,15,9,9,14,8,6,6
   1260 DATA 20,15,14,12,10,8,6,6
   1270 FOR J=2 TO 4
   1280 FOR I=1 TO 8
   1290 READ F(J,I)
   1300 NEXT I
   1310 NEXT J
   1320 DATA 5,4,3,3,2,2,1,1
   1330 DATA 5,4,3,1,2,1,3,1
   1340 DATA 4,3,2,2,3,1,1,1
   1350 FOR I=1 TO D*3
   1360 READ P(I)
   1370 NEXT I
   1380 DATA "STRENGTH","VITALITY","AGILITY","INTELLIGENCE"
   1390 DATA "EXPERIENCE","LUCK","AURA","MORALITY","CHARACTER CREATION"
   1400 DATA "2 HAND SWORD","BROADSWORD","SHORTSWORD"
   1410 DATA "AXE","MACE","FLAIL","DAGGER","GAUNTLET","ARMOURY"
   1420 DATA "HEAVY ARMOUR","CHAIN ARMOUR","LEATHER ARMOUR","HEAVY ROBE"
   1430 DATA "GOLD HELMET","HEADPIECE","SHIELD","TORCH","ACCOUTREMENTS"
   1440 DATA "NECRONOMICON","SCROLLS","RING","MYSTIC AMULET","SASH","CLOAK"
   1450 DATA "HEALING SALVE","POTIONS","EMPORIUM"
   1460 FOR J=1 TO 4
```

Initialization starts here. Lines 1080-1130 dimension the arrays.

These are the flags. They are explained on page 18.

Here's the array holding the names of attributes and all the things that you can buy.

```
  1470 FOR I=1 TO 9
  1480 READ F$(J,I)
  1490 NEXT I
  1500 NEXT J
  1510 DATA "WANDERER","CLERIC","MAGE","WARRIOR","BARBARIAN"
  1520 FOR I=1 TO 5
  1530 READ C$(I)
  1540 NEXT I
☆ 1550 LET MP=3+rnd(5)
☆ 1560 LET GC=120+rnd(60)
  1570 LET M$="":LET AS=65
  1580 LET B$="":FOR I=1 TO W:LET B$=B$+" ":NEXT I
  1590 RETURN
▲○■★ 1600 REM SET UP DETAILS
▲○■★ Add whichever lines between 1610 and 1730 are indicated in the conversions for your computer.
▲■ Add lines 4000-4030 from page 35.
```


Here are the names of the character-types you can create.

Conversions

In these boxes you will find the conversion lines you must use in the character creator listing. Make sure you pick the right ones for your computer.

BBC/Electron ★

```
1600 MODE 5:VDU 23,0,8202;0;0;0;
1610 W=20:B=224
1620 VDU 23,224,255,255,255,255,255,255,255,255
1630 RETURN
```

Commodore 64/Vic ■▲

The lines below are the same for the C64 and Vic listings.

```
  20 PRINT CHR$(147)
  90,310 Replace ";" with "+"
 380 X=1:Y=3:GOSUB 1700:N$=IN$
 530 OPEN 1,1,1,"HERO"
 540 PRINT#1,S$
 550 CLOSE 1
 590 X=14:Y=2:GOSUB 1700:OF=VAL(IN$)
 875 PRINT HM$;LEFT$(CU$,3);SPC(15);LEFT$(B$,4);
 945 PRINT HM$;LEFT$(CU$,Y);SPC(15);LEFT$(B$,5);
1020 PRINT BG$(FG);" ";BG$(BG);LEFT$(B$,W-2);BG$(FG);" ";
1610 GOSUB 4000
1650 RETURN
1700 IN$=""
1710 GET I$:IF I$=CHR$(13) THEN RETURN
1720 IF I$>"/" AND I$<"[" THEN LET IN$=IN$+I$:PRINT HM$;LEFT$(CU$,Y);SPC(X);IN$;
1730 GOTO 1710
4000-4030 Add these lines from page 35.
```

These two lines must be added as well on the C64.■

```
  25 POKE 53280,0:POKE 53281,0
1600 LET W=40
```

If you have a Vic then these two lines must be used, too. ▲

```
  25 POKE 36879,8
1600 LET W=22
```

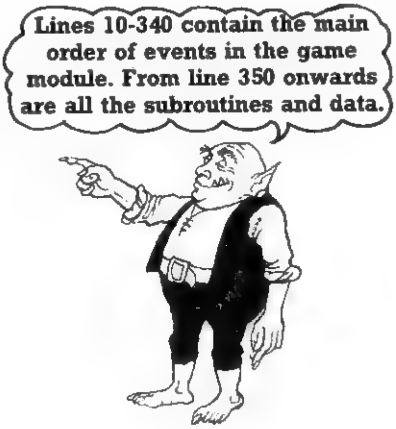
Spectrum ○

```
  20 BORDER 0:CLS
  90,310 Replace ";" with "K"
 100,320 Replace "-" with "J"
370,820,870,990,1020,1040 Replace LEFT$
   (B$,exp) with B$( TO exp)
 530 LET Q$(1)=S$
 540 SAVE "HERO" DATA Q$()
```

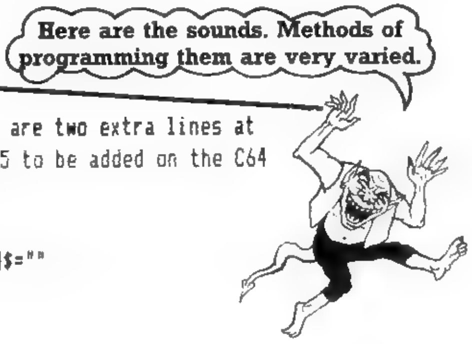
```
 550 Leave out this line
 690 IF O$(N)(C)="1" THEN LET Y=1
1090 DIM F$(4,D+1,18)
1100 DIM C$(5,9)
1120 DIM O$(D+3,5):DIM Q$(1,57)
1600 LET B=143:LET W=32
1610 DIM C(4):LET C(1)=0:LET C(2)=2:
     LET C(3)=6:LET C(4)=7
1620 RETURN
```

Game Module

The Game Module listing starts here. It is very long so you must be very accurate. Take a break if you get tired. Make sure you include all the conversion lines marked with the symbol for your computer (see symbols key on page 32).


Lines 10-340 contain the main order of events in the game module. From line 350 onwards are all the subroutines and data.

```
▲ See special note for Vic users page 46.
■  10 GOSUB2500
   20 GOSUB2010
   30 GOSUB1770
☆  40 LET I$=inkey$
   50 IF I$="A" AND DX<255 THEN GOSUB870
   60 IF I$="C" AND F(7)>0 AND O(17)+O(18)>0 THEN GOSUB990
   70 IF I$="G" THEN GOSUB1410
   80 IF I$="P" THEN GOSUB1660
   90 IF I$="R" THEN GOSUB1690
  100 IF I$="S" THEN GOSUB2260
  110 IF I$="B" THEN LET NF=NF-1
  120 IF I$="N" THEN LET NF=NF+1
  130 IF NF>4 THEN LET NF=1
  140 IF NF<1 THEN LET NF=4
  150 IF I$="M" THEN LET NX=NX+D(NF,1): LET NY=NY+D(NF,2)
  160 IF NY>15 THEN LET NY=15
  170 IF NY<1 THEN LET NY=1
  180 IF NX<1 THEN LET NX=1
  190 IF NX>15 THEN LET NX=15
  200 LET RH=R(NX,NY)
■ 210 IF RH=C1 THEN LET X=NX:LET Y=NY:GOSUB570:LET NX=OX:LET NY=OY:LET F(1)=F(1)-.03
  220 IF RH=C6 THEN LET TX=NX:LET TY=NY:LET TF=1
  230 IF TF=1 THEN LET NX=TX:LET NY=TY
☆ 240 IF F(1) > S1*.8 AND rnd(8)<F(6) THEN LET TF=0
  250 IF I$>""THEN LET F(1)=F(1)*0.99
  260 IF F(1) <S1 THEN LET F(1)=F(1)+(F(2)/1100)
  270 GOSUB480
  280 IF OX<>NX OR OY<>NY THEN LET X=OX:LET Y=OY:GOSUB570
  290 LET OX=NX:LET OY=NY
  300 IF DX<255 THEN GOSUB620
  310 IF F(1)>0 AND FI<1 AND RH<>C5 THEN GOTO40
  320 IF RH=C5 THEN LET M$=T$(12):GOSUB430:GOSUB1760:GOTO40
  330 IF F(1)<1 THEN GOSUB810
☆ 340 PRINT tab(0,10);:STOP
■▲350 sound a note :RETURN
○★360 sound a noise :RETURN
☆ 370 paper 2:ink 0
☆ 380 PRINT tab(0,5);M$;
☆ 390 LET I$=inkey$
  400 IF I$="" THEN GOTO390
○☆410 PRINT tab(0,5);LEFT$(B$,W);:LET M$=""
  420 RETURN
☆ 430 paper 2:ink 0
☆ 440 PRINT tab(0,5);M$;
○■▲450 FOR D=1 TO 600:NEXT D
☆○460 PRINT tab(0,5);LEFT$(B$,W);:LET M$=""
```


Here are the sounds. Methods of programming them are very varied.

NOTE:There are two extra lines at 355 and 365 to be added on the C64 and Vic.

These lines print your character's attributes on the screen. Use the conversion charts for colour and print commands very carefully. They are on page 32.

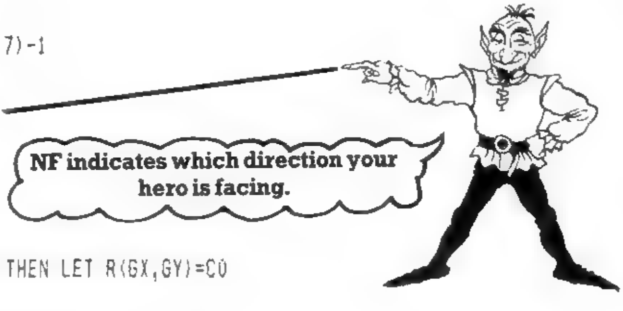
```
     470 RETURN
    ☆480 paper 1:ink 3
   ○☆490 PRINT tab(NX,NY+5);MID$(F$,NF,1);
  ▲■☆500 paper 2:ink 0
  ▲■☆510 PRINT tab(16,8);INT(F(1));" ";
  ▲■☆520 PRINT tab(16,11);INT(F(2));" ";
     530 PRINT tab(16,14);INT (F(7));" ";
   ○☆540 PRINT tab(16,17);MID$("NESW.",NF,1);
    ☆550 PRINT tab(16,20);INT(F(5));
     560 RETURN
    ☆570 paper 1:ink 2
    ☆580 LET RM=R(X,Y):PRINT tab(X,Y+5);CHR$(RM);
     590 IF ABS(DX)<4 OR RM<=C7 THEN RETURN
     600 LET MT=RM:LET M=MT-C2:LET MV=M/16:LET MS=M*6:LET DX=3:LET LX=X:LET LY=Y
     610 RETURN
     620 LET DX=LX-NX:LET SX=SGN(DX):LET DY=LY-NY:LET SY=SGN(DY)
     630 LET MX=LX-(MV*SX):LET MY=LY-(MV*SY):LET RM=R(MX,MY)
     640 IF RM>C0 AND RM<>MT THEN LET MY=LY:LET MX=LX
     650 LET R(LX,LY)=C0:LET X=LX:LET Y=LY:GOSUB 570
     660 LET R(MX,MY)=MT:LET X=MX:LET Y=MY:GOSUB 570
     670 LET LX=MX:LET LY=MY:LET H=0
     680 IF ABS(DX)<=1 AND ABS(DY)<=1 AND RH<>C7 THEN LET H=M*.5:LET J=H:GOSUB350
     690 IF H*12<F(6)+F(3) THEN RETURN
     700 LET M$=T$(5):GOSUB430:GOSUB360
     710 LET H=H/(3+O(9) + O(10) + O(11) + O(12) + O(13) + O(14))
     720 LET F(1)=F(1)-H:LET F(2)=F(2)-(H/101)
    ☆730 LET I=1:LET WB=0:LET MB=rnd(M)
     740 LET J=MT:GOSUB350:GOSUB360
     750 IF MB=1 AND O(I)>0 THEN GOSUB780
     760 IF I<11 THEN LET I=I+1:GOTO750
     770 RETURN
     780 LET O(I)=0:LET M$=T$(8)+" "+W$(I):GOSUB430
     790 LET MB=0:GOSUB360:LET J=I:GOSUB350
     800 RETURN
     810 LET NF=5:LET F(1)=0:GOSUB440
    ☆820 PRINT tab(1,5);"THOU HAST EXPIRED!"
     830 FOR J=150 TO 1 STEP-4
     840 GOSUB350:GOSUB360:GOSUB570:GOSUB480
     850 NEXT J
     860 RETURN
    ☆870 LET M$=T$(rnd(3)):GOSUB360
    ☆880 LET H=F(1)+O(1) + O(2) + O(3) + O(4) +O(5) + O(7) + O(8)+rnd(F(6))
    ☆890 IF F(3)+F(6)< rnd(M)+2 THEN LET M$=T$(4):LET HT=0
     900 LET MS=MS-H:GOSUB430
     910 LET F(1)=F(1)-(H/100):LET F(5)=F(5)+.05
     920 IF MS<1 THEN GOSUB940
     930 RETURN
     940 LET DX=255:LET MS=0:LET R(MX,MY)=C0
     950 LET F(5)=F(5)+.1
```

These are the sound subroutines. Lines like this give instructions for how the sound should be produced.

```
  960 LET M$=T$(6):GOSUB430
  970 FOR J=200 TO 150STEP-8:GOSUB350:GOSUB360:NEXT J
  980 GOSUB570:RETURN
☆ 990 GOSUB480:paper 2:ink 0
☆1000 PRINT tab(0,1);"YOU MAY USE MAGICKS";
☆1010 IF O(17)>0 THEN PRINT tab(0,2);"FROM NECRONOMICON";
☆1020 IF O(18)>0 THEN PRINT tab(0,3);"FROM THE SCROLLS";
☆1030 PRINT tab(0,4);"CONSULT THE LORE"
 1040 LET M$="USE SPELL NUMBER?":GOSUB370
○1050 LET SL=VAL(I$)
 1060 IF SL=0 OR (O(17)=0 AND SL<5) OR (O(18)=0 AND SL>3) OR SL>6 THEN GOTO1040
 1070 LET M(SL)=M(SL)-1:LET X=NX:LET Y=NY
 1080 IF M(SL)<0 THEN LET M$=T$(9):LET SL=7
▲■
○☆1090 FOR J=1 TO 5:PRINT tab(0,J);LEFT$(B$,W);:NEXT J:GOSUB570
○1100 ON SL GOSUB1140,1190,1220,1280,1300,1390,1130
 1110 LET F(5)=F(5)+.2
 1120 GOSUB430
 1130 RETURN
 1140 FOR J=1 TO 12
 1150 GOSUB350:GOSUB360
 1160 NEXT J
 1170 IF DX<255 THEN LET X=MX:LET Y=MY:GOSUB940
 1180 RETURN
 1190 IF RH=C0 THEN LET R(NX,NY)=C7
 1200 LET J=100:GOSUB350:LET J=200:GOSUB350
 1210 RETURN
☆1220 LET NX=rnd(13):LET NY=rnd(13)
 1230 FOR J=0 TO 255 STEP8
 1240 GOSUB360:GOSUB350
 1250 NEXT J
 1260 GOSUB480
 1270 RETURN
▲■☆1280 LET F(2)=F(2)+rnd(M(SL)):LET F(1)=F(1)+rnd(M(SL)):LET F(7)=F(7)-1
 1290 RETURN
 1300 FOR J=1 TO 30
☆1310 LET R(NX,NY)=rnd(8)+1+C0
 1320 GOSUB350:GOSUB570
 1330 NEXT J
 1340 IF RH<=C7 THEN LET DX=255:LET MS=0
 1350 FOR J = 1 TO 20 STEP4
 1360 GOSUB 350
 1370 NEXT J
 1380 RETURN
 1390 LET F(2)=S2:LET F(1)=S1:LET F(7)=F(7)-1
 1400 RETURN
 1410 LET GX=NX+D(NF,1):LET GY=NY+D(NF,2)
 1420 IF GX<0 THEN LET GX=0
 1430 IF GY<0 THEN LET GY=0
 1440 IF GX>15 THEN LET GX=15
 1450 IF GY>15 THEN LET GY=15
 1460 LET GT=R(GX,GY):IF GT>C1 AND GT<C4 THEN LET R(GX,GY)=C0
```

Here's where the magic routines start.

All the spell routines are on this page. Look out for all the random commands(rnd).

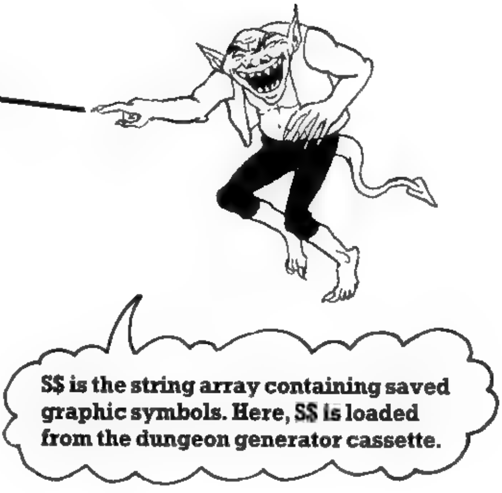

NF indicates which direction your hero is facing.

```
  1470 IF GT=C2 THEN LET O(23)=O(23)+1:LET O(24)=O(24)+1
  1480 IF GT=C3 THEN LET TR=TR+1
  1490 IF GT=C4 THEN GOSUB 1550
  1500 LET X=GX:LET Y=GY:GOSUB570
  1510 IF GT>C1 AND GT<C4 THEN LET J=GT:GOSUB350:LET J=GT+5:GOSUB350
  1520 RETURN
  1530 LET J=O(24)+O(23)*10:GOSUB350
  1540 RETURN
 ☆1550 paper 2:ink 1
 ☆1560 PRINT tab(0,1);" THY QUEST IS OVER! "
  1570 FOR I = 1 TO 18
  1580 LET J=T(I):GOSUB350
  1590 LET X=NX:LET Y=NY
  1600 FOR N=1 TO 4:LET NF=N:GOSUB480:NEXT N
  1610 NEXT I
  1620 LET MS=0
 ☆1630 PRINT tab(1,2);"THY SCORE=";INT((TR*10)+(GC*F(5))+F(1)+F(2)+F(3))
  1640 LET FI=1
  1650 RETURN
  1660 IF O(24)>0 AND F(1)<S1 THEN LET F(1)=S1:LET O(24)=O(24)-1
  1670 IF O(23)>0 AND F(2)<S2 THEN LET F(2)=S2:LET O(23)=O(23)-1
  1680 RETURN
  1690 IF LT=0 THEN LET M$=T$(7):GOSUB430:RETURN
  1700 FOR Y=NY-3 TO NY+3
  1710 FOR X=NX-3 TO NX+3
  1720 IF (X>0 AND X<16) AND (Y>0 AND Y<16)THEN GOSUB570
  1730 NEXT X:NEXT Y
  1740 LET LT=LT-1
  1750 RETURN
  1760 IF F(5)<S3+1 THEN LET M$=T$(11):LET NX=OX:LET NY=OY:GOSUB430:RETURN
▲■☆1770 CLS:PRINT tab(0,3);"PREPARE DUNGEON TAPE"
  1780 LET M$=T$(10):GOSUB370
○▲■1790 S=OPENIN"LEVEL"
  ▲■
○▲■1800 INPUT#S,S$
  ▲■
○▲■1810 CLOSE#S
  1820 LET I=1
  1830 FOR Y=1 TO 15
  1840 FOR X=1 TO 15
 ○1850 LET R(X,Y)=ASC(MID$(S$,I,1))
  1860 LET I=I+1
  1870 NEXT X
  1880 NEXT Y
 ○1890 LET IX=ASC(MID$(S$,I,1))-OS
 ○1900 LET IY=ASC(MID$(S$,I+1,1))-OS
 ○1910 LET LE=ASC(MID$(S$,I+2,1))-OS
  1920 IF LE>F(5) THEN GOSUB 1960:GOTO 1760
  1930 GOSUB2790
  1940 LET NX=IX:LET NY=IY:LET OX=NX:LET OY=NY:LET DX=255
  1950 RETURN
  1960 PRINT:PRINT"LEVEL TOO DEEP"
  1970 PRINT"REWIND TAPE"
```

These lines are used when the Idol has been found in the last dungeon and the Quest is over. Line 1630 calculates your final score by adding together treasure, experience, strength, vitality and agility scores. You can try to improve your score next time you play.

You're over half-way through the game module listing now, so don't give up.

```
   1980 PRINT"TO POSITION"
   1990 PRINT"FOR LEVEL";F(5)
   2000 RETURN
▲■☆2010 CLS:PRINT tab(0,3);"PREPARE HERO TAPE"
   2020 LET M$=T$(10):GOSUB370
○▲■2030 S=OPENIN "HERO"
○▲■2040 INPUT#S,S$
○▲■2050 CLOSE#S
   2060 LET P=2
  ○ 2070 LET OT=ASC(MID$(S$,1,1))-AS
   2080 FOR I= 1 TO 8
  ○ 2090 LET F(I)=ASC(MID$(S$,P,1))-AS
   2100 LET P=P+1
   2110 NEXT I
   2120 FOR I=1 TO OT
  ○ 2130 LET D(I)=ASC(MID$(S$,P,1))-AS
   2140 LET P=P+1
   2150 NEXT I
  ○ 2160 LET GC=ASC(MID$(S$,P,1))-AS
  ○ 2170 LET TR=ASC(MID$(S$,P+1,1))-AS
  ○ 2180 LET C$=RIGHT$(S$,LEN(S$)-(P+1))
   2190 LET S1=F(1):LET S2=F(2):LET S3=F(5)
   2200 FOR I=1 TO 2
   2210 FOR J=1 TO 3
   2220 LET M((I-1)*3+J)=D(16+I)*F(7)
   2230 NEXT J:NEXT I
   2240 IF D(16)=1 THEN LET LT=20
   2250 RETURN
   2260 LET M$="ONE MOMENT PLEASE":GOSUB430
   2270 LET S$="":LET T$=""
   2280 FOR Y=1 TO 15
   2290 FOR X=1 TO 15
   2300 LET T$=T$+CHR$(R(X,Y))
   2310 NEXT X:NEXT Y
   2320 LET T$=T$+CHR$(OS+NX)
   2330 LET T$=T$+CHR$(OS+NY)
   2340 LET T$=T$+CHR$(OS+LE)
   2350 LET S$=S$+CHR$(AS+OT)
   2360 FOR I=1 TO 8
   2370 LET S$=S$+CHR$(F(I)+AS)
   2380 NEXT I
   2390 FOR I=1 TO OT
   2400 LET S$=S$+CHR$(D(I)+AS)
   2410 NEXT I
   2420 LET S$=S$+CHR$(GC+AS)
   2430 LET S$=S$+CHR$(TR+AS)
   2440 LET S$=S$+C$
   2450 LET M$="ANY KEY TO SAVE":GOSUB370
○▲■2460 S=OPENOUT"HERO":PRINT#S,S$:CLOSE#S
○▲■2470 S=OPENOUT"LEVEL":PRINT#S,T$:CLOSE#S
 ▲■2480 LET FI=1
```


S$ is the string array containing saved graphic symbols. Here, S$ is loaded from the dungeon generator cassette.

```
   2490 RETURN
   2500 LET C$="ROLE PLAYING GAME":LET B$=""
★○▲2510 LET W=40:LET OS=96
   2520 FOR I=1 TO W:LET B$=B$+" ":NEXT I
  ○ 2530 DIM R(15,15),F(8),D(24)
  ○ 2540 DIM W$(11),T$(12)
 ○○ 2550 DIM M(6),D(4,2),T(18)
   2560 DATA"GR SWORD","SWORD","AXE","MACE",
        "FLAIL","DAGGER","ARMOUR","ARMOUR"
   2570 DATA"ARMOUR","HELMET","HEADPC."
   2580 FOR I = 1 TO 11
   2590 READ W$(I)
   2600 NEXT I
   2610 DATA"A GOOD BLOW","WELL HIT SIRE",
        "THY AIM IS TRUE","MISSED!","HIT THEE!!"
   2620 DATA"THE MONSTER IS SLAIN","NO LIGHT",
        "BROKEN THY ","SPELL EXHAUSTED"
   2630 DATA"PRESS ANY KEY","YOU NEED EXPERIENCE",
        "EXIT FROM THIS LEVEL"
   2640 FOR I = 1 TO 12
   2650 READ T$(I)
   2660 NEXT I
   2670 DATA0,-1,1,0,0,1,-1,0
   2680 FOR I=1 TO 4:READ D(I,1),D(I,2):NEXT I
   2690 LET FI=0:LET DX=255:LET NF=0
   2700 LET TX=0:LET TY=0:LET TF=0:LET TR=0
   2710 LET MX=0:LET MY=0:LET DY=12:LET F$=""
   2720 LET NX=1:LET NY=1:LET RE=0:LET LT=0
   2730 FOR I = 1 TO 5
   2740 LET F$=F$+CHR$(OS+I)
   2750 NEXT I
   2760 DATA69,117,73,121,81,129,69,117,73,121,
        81,129,89,137,97,145,101,149
   2770 FOR I = 1 TO 18:READ T(I):NEXT I:
        GOSUB 2930
   2780 RETURN
■▲☆2790 paper 1:CLS
■■▲
▲■☆2800 paper 3:ink 0
  ○2810 PRINT C$;LEFT$(B$,W-LEN(C$));
  ☆2820 paper 2:ink 3
  ○2830 FOR I = 1 TO 5:PRINT LEFT$(B$,W);:
        NEXT I
▲■☆2840 paper 0:ink 1
 ○☆2850 FOR I=1 TO 15:PRINT tab(1,5+I);
        LEFT$(B$,15);:NEXT I
  ☆2860 paper 1:ink 3
  ☆2870 PRINT tab(16,7);"STR";
  ☆2880 PRINT tab(16,10);"VIT";
  ☆2890 PRINT tab(16,13);"AUR";
  ☆2900 PRINT tab(16,16);"FACE";
  ☆2910 PRINT tab(16,19);"EXP";
  ▲
   2920 RETURN
   2930 REM *** USER DEF'D CHARACTERS ***
```

■▲○★Add whichever lines between 2940 and 3000 are indicated in the conversions for your computer.

```
3170 LET AS=65:LET C0=OS+6
3180 LET C1=C0+1:LET C2=C0+2:LET C3=C0+3:
     ,LET C4=C0+4
3190 LET C5=C0+6:LET C6=C0+7:LET C7=C0+8:
     LET C8=C0+12
3200 RETURN
3500 DATA 255,255,255,255,255,255,255,255
3510 DATA 56,56,16,56,84,16,40,68
3520 DATA 28,28,8,30,40,40,20,18
3530 DATA 56,56,16,84,56,40,68,40
3540 DATA 56,56,16,120,148,20,40,72
3550 DATA 0,0,0,0,8,209,254,221
3560 DATA 0,0,0,0,0,0,0,0
3570 DATA 85,170,85,170,85,170,85,170
3580 DATA 0,60,24,60,126,126,126,60
3590 DATA 0,56,100,114,95,73,41,31
3600 DATA 20,42,20,20,93,93,62,99
3610 DATA 60,126,255,255,255,253,255,255
3620 DATA 60,102,195,129,129,129,133,129
3630 DATA 129,66,36,0,0,36,66,129
3640 DATA 0,60,66,66,66,66,60,0
3650 DATA 76,158,170,190,84,30,37,88
3660 DATA 0,56,84,124,56,44,68,102
3670 DATA 0,8,28,42,127,85,65,34
```

▲■ Add lines 4000-4030 from page 35 but this time replace CHR$(158) with CHR$(31) in line 4010

■ Add lines 5000-5080 from page 35.

Conversions

Spectrum ○

```
 350 BEEP 0.05,J/2-60:RETURN
 360 BEEP 0.05,J/2-60:RETURN
 410,460,1090,2810,2830,2850 Replace LEFT$
 (B$,exp) with B$(TO exp)
 450 FOR D=1 TO 50:NEXT D
 490 PRINT AT NY+5,NX;F$(NF);
 540 PRINT AT 17,16;"NESW."(NF)
1045 IF I$<"1" OR I$>"6" THEN GOTO 1040
1100 GOSUB 1140*(SL=1)+1190*(SL=2)+1220*
     (SL=3)+1280*(SL=4)+1300*(SL=5)
     +1390*(SL=6)+1130*(SL=7)
1790 LOAD "LEVEL" DATA Q$()
1800 LET S$=Q$(1)
1810 Leave out this line
1850 LET R(X,Y)=CODE(S$(I))
1890 LET IX=CODE(S$(I))-OS
1900 LET IY=CODE(S$(I+1))-OS
1910 LET LE=CODE(S$(I+2))-OS
2030 LOAD "HERO" DATA Q$()
2040 LET S$=Q$(1)
2050 Leave out this line
2070 LET OT=CODE(S$(1))-AS
2090 LET F(I)=CODE(S$(P))-AS
2130 LET O(I)=CODE(S$(P))-AS
2160 LET GC=CODE(S$(P))-AS
2170 LET TR=CODE(S$(P+1))-AS
2180 LET C$=S$(P+2 TO )
2460 LET K$(1)=S$:SAVE "HERO" DATA K$()
2470 LET L$(1)=T$:SAVE "LEVEL" DATA L$()
2510 LET W=32:LET OS=144
2530 DIM R(15,15):DIM F(8):DIM O(24)
2540 DIM W$(11,8):DIM T$(12,20)
2545 DIM K$(1,57):DIM L$(1,228)
2550 DIM M(6):DIM D(4,2):DIM T(18)
2950 FOR I=0 TO 143:READ A:POKE USR "A"+I,
     A:NEXT I
2960 DIM C(4):LET C(1)=0:LET C(2)=2:LET
     C(3)=6:LET C(4)=7
```

BBC/Electron ★

```
 350  SOUND 1,-15,J,2:RETURN
 360  SOUND 0,-15,7,3:RETURN
2510  LET W=20:LET OS=224
2940  MODE 5:VDU 23;8202;0;0;0;
2960  FOR I=0 TO 17:VDU 23,224+I
2970  FOR J=0 TO 7:READ A:VDU A
2980  NEXT J:NEXT I
```

Commodore 64/Vic ■▲

```
 500  PRINT BG$(1);
 510,520,530 At the end of these lines
 replace " "; with CHR$(157);" ";
1085  PRINT BG$(2);
210,1280 Leave out the word LET in these lines
1770 ,2010 Replace CLS with PRINT CHR$(147)
1790  OPEN 1,1,0,"LEVEL"
1795  S$="":FOR I=1 TO 3
1800  INPUT#1,TP$:S$=S$+TP$
1805  NEXT I
1810  CLOSE 1
2030  OPEN 1,1,0,"HERO"
2040  INPUT#1,S$
2050  CLOSE 1
2460  OPEN 1,1,1,"HERO":PRINT#1,S$:CLOSE 1
2465  OPEN 1,1,1,"LEVEL":FOR I=0 TO 2
2470  PRINT#1,MID$(T$,I*76+1,76)
2475  NEXT I:CLOSE 1
2800  PRINT HM$;CHR$(144);BG$(0);
2840  PRINT BG$(1);CHR$(144);
2940  GOSUB 4000
4010  Replace CHR$(158) with CHR$(31)
4000-4030    Add these lines from page 35.
```

Vic ONLY ▲ *SPECIAL NOTE FOR VIC USERS*

Before typing in or loading this program type in the following: POKE 44,28:POKE 642,28:SYS(64824)

```
 350  POKE 36876,J/2+128:FOR TT=1 TO 5:NEXT TT
 355  POKE 36876,0:RETURN
 360  POKE 36877,J/2+128:FOR TT=1 TO 5:NEXT TT
 365  POKE 36877,0:RETURN
 450  FOR D=1 TO 200:NEXT D
2510  LET W=22:LET OS=96
2790  POKE 36879,24:PRINT BG$(1);
2795  PRINT CHR$(147);:FOR I=0 TO 504:PRINT CHR$(32);
      :NEXT I
2915  POKE 4601,160:POKE 38393,2
2950  FOR I=0 TO 2047:POKE 5120+I,PEEK(32768+I):NEXT I
2960  FOR I=0 TO 143:READ A:POKE 6656+I,255-A:NEXT I
2970  POKE 36869,205:POKE 36878,15
```

Line changes for additional graphics:
When you add a new user-defined symbol you must change these lines in the dungeon generator and game module:

```
                   DUNGEON GENERATOR
   C64       830   FOR I=0 TO 95: ←
   VIC       840   Add 8 to this number
   SPECTRUM  830   for each new symbol
   BBC       830   FOR I=0 TO 11: ←
                   Add 1 to this number
                   for each new symbol.
                   GAME MODULE
   C64      2950   FOR I=0 TO 43: ←
   VIC      2960   Add 8 to this number
   SPECTRUM 2950   for each new symbol.
   BBC      2960   FOR I=0 TO 17: ←
                   Add 1 to this number
                   for each new symbol.
```

Phew! Finished at last.

C64 ONLY ■

```
   5 GOSUB 5000:POKE 53281,0
 350  POKE VS+1,J:POKE VS+4,33
 355  POKE VS+4,32:RETURN
 360  POKE VS+8,J:POKE VS+11,129
 365  POKE VS+11,128:RETURN
 450  FOR D=1 TO 200:NEXT D
2790 POKE 53280,0:POKE POKE 53281,1:
      PRINT BG$(1);
2795  PRINT CHR$(147);:FOR I=0 TO
      998:PRINT CHR$(32);:NEXT I
2797  POKE 33767,160:POKE 56295,2
2950  FOR I=0 TO 143:READ A:
      POKE 36352+I,255-A:NEXT I
2960  VS=54272
2970  POKE VS+24,15
2980  POKE VS+5,9:POKE VS+6,0
2990  POKE VS+12,9:POKE VS+13,0
3000  POKE VS+7,0:POKE VS,0
5000-5080 Add these lines
          from page 35.
```

INDEX

American Standard Code for Information Interchange 16
array 12, 14, 15, 18
ASCII 16, 19
attributes 10, 18, 22
  changing 6, 9
  scores 7, 11, 15, 27, 31

bargaining 8, 19
BBC micro 3, 20
Book of Lore 28, 29, 30, 31
bugs 24

CO (meaning of) 16, 20, 25, 26
character 4, 5, 6
  creator 10, 14, 18, 19, 29
  loading into game 19, 30
  saving 19, 29
  types 4, 6, 7, 8, 10, 29
Commodore 64 3, 20
conjure 21
conversions 34, 35, 39, 45, 46
  general chart 32

data 12, 14-16
de-bugging 24
dimensioning arrays 15, 27
dungeon 5
  generator 10, 14, 16, 17, 25, 27
  levels 6, 9, 11, 28
  loading into the game 17, 30
  master 8, 9
Dungeon of Doom 3,-6, 8, 10, 12
  program listing 32-46
  rules 28-31
  story of 3
  expanding 25
Dungeons and Dragons 4

Electron 3, 20

F, F$ 18
fantasy games 3, 4, 5,
flags 18

game module 10, 11, 25, 26, 30
  loading character into 19
  loading dungeon into 17
  what it does 12, 20
  routines in 22, 23
graphics
  in Dungeon of Doom game 28
  new torch symbol 26
  user-defined 5, 16

hero 4, 6

idol 3, 28, 31
initialization 13, 14, 15, 16, 18

listings 33-45
load
  dungeon 17, 30
  character 19, 30
  game 30

magic 21, 31
  items 8, 27
messages 12, 14, 15, 29, 30
  secret 26, 27
mis-loading 24
modification points 6, 7, 10, 29

Necronomicon 8, 27, 31

offset numbers 16, 19

program
  listings 33-46

Quest 4, 30, 31

random command 25
role-playing game 4
RPG 4
routines 11
  character creator 18, 19
  dungeon generator 16, 17
  game module 22, 23
rules 3, 28-31

$ sign 14
save
  characters 10, 29
  dungeons 10, 17, 28
  game 11, 23, 31
scores 7, 11, 15, 27, 31
screen-handling 20
secret
  messages 26, 27
  squares 26
solo game 5
sounds 20
Spectrum 3, 20
string 14
  variable 17
subroutine 12, 13, 15
  action 20, 22, 23
  conjure 21
  service 20

torch symbol 26
trading post 8, 9, 27
TSR Hobbies Inc, 4

user-defined graphics 5, 16

VAL 16, 21
variable 17, 20
Vic 3, 20

Answers

Page 14.
M$(3) is the message "Thy aim is true".

The goblin is putting the monster in location R(14,14).

There is a potion in location R(15,12).

Page 18
Here are the "translations" of the lines that limit character-type:
IF F(1,4)>8 AND F(1,7)>7 THEN LET C=3
If the intelligence score is more than eight and the aura is more than seven, then the character-type will be a mage.

IF F(1,1)>7 AND F(1,8)>5 AND F(1,1)+F(1,2)>10 THEN LET C=4
If the strength score is more than seven and the morality score is less than five, and strength plus vitality equals more than ten, then the character-type will be a warrior.

IF F(1,1)>8 AND F(1,2)+F(1,3)>12 AND F(1,8) <6 THEN LET C=5
If the strength score is more than eight and vitality plus agility is more than twelve, and morality is less than six, then the character-type will be a barbarian.

Fantasy games notice board

If you've really got the fantasy game itch now and want to find out more about them, the notices on these pages will give you some ideas.

Computer fantasy games

If you enjoyed playing *Dungeon of Doom* you will probably be interested in other fantasy games to play on your computer. There are lots available in the shops, but read the description carefully before you buy as they don't all include role-playing, which is the main difference between a fantasy and an adventure. Here are some of the role-playing games you should be able to find:

The Hobbit by Melbourne House for Spectrum 48K, Commodore 64 and BBC (BBC version is text-only).
Valhalla by Wintersoft for Spectrum 48K and soon for Commodore 64.
Ring of Darkness by Wintersoft for Spectrum 48K.
Sherlock Holmes by Melbourne House. Available soon.

Magazines for micro-users

The best place to find out about computer games is in a computer magazine. Most magazines for micro-users include reviews of new games and some, like *Games Computing* and *Personal Computer Games* specialize in articles on all aspects of computer gaming. There is even a magazine devoted entirely to adventure and fantasy games. It is called *Micro-Adventurer* and has news, critical reviews, articles and program lines that you can add to games you have written. It also has a very useful "Adventure Help" page through which you can ask other readers' advice if you have a problem with a particular game.

Fantasy board games

You may be inspired by games you play on your computer to try the non-computer variety. There are lots to choose from, using all sorts of settings. Most toy shops should have a selection. Here are just a few of those available:
Dungeons and Dragons by TSR (the first fantasy role-playing game produced).
Freedom in the Galaxy by Avalon Hill (a space fantasy).
Gangbusters by TSR (1920s role-playing).
Runequest by Chaosium (mythical world with monsters and magic).
Champions by Hero Games (super-heroes with super-human abilities).

Magazines for fantasy fans

Just as in the case of computer fantasy games, the best place to discover more about fantasy board games is in specialist magazines. An American magazine called *The Dragon*, which is also available in English newsagents, is published by TSR, the makers of *Dungeons and Dragons*, and is all about aspects of that game. *Imagine* is published in England by TSR UK Ltd and caters for players of all sorts of role-playing games, although *Dungeons and Dragons* is featured strongly. If you buy just one copy of either of these magazines you will find addresses of game-stockists and mail-order houses, role-playing scenarios you can use and information on games, books and films that the fantasy gamer might enjoy.

There are also smaller, less well-known magazines that cater for other fantasy games. Their circulation is more specialized so they are not generally available through newsagents, but can be obtained through mail-order houses.

Books, films and television

If you have difficulty thinking of a theme for your fantasy game, you should be able to find some inspiration in books, films or television. There are already fantasy games based on books, like The Hobbit, and board games based on television programs, such as Doctor Who. The game Champions is derived from comic strips. You could research a period in history or base a futuristic game on your favourite piece of science fiction.

Postal games

If you can't find other people to play with, magazines can help you again. They advertise postal games, such as *Crasimoff's World*, which anyone can join. You pay an enrolment fee to start and usually pay for each round you play, too. You decide what a group of characters should do, given the situation described, and in the next round you are told the consequences of their actions and the new situation to work from.

Programming books

If you want to learn more about BASIC, which is the computer language used in *Dungeon of Doom*, here are some Usborne books that will help you:
"Introduction to Computer Programming" by B. Reffin-Smith
"Better Basic" by B. Reffin-Smith and L. Watts
"Practise Your Basic" by N. Cutler
"Programming Tricks and Skills" by L. Watts and L. Howarth

Clubs and events

To meet other people who enjoy playing fantasy games, you can join a club, or place a personal advertisement in one of the magazines mentioned opposite.

Some people want to act out their fantasy roles for real, and now they can do so by joining special clubs. These lease a location, such as a castle, and populate it with fantasy monsters (club members in disguise). You form a group with other members and try to find treasure and come out "alive".

Conventions and games fairs that include role- playing games, are sure to be announced in fantasy game magazines.

First published in 1984 by Usborne
Usborne Publishing Ltd,
20 Garrick Street,
London WC2E 9BJ,
England.

Copyright © 1984 Usborne
Publishing

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher.

The name Usborne and the device 🎈 are Trade Marks of Usborne Publishing Ltd.

Printed in Spain

Usborne Computer Books

"Highly recommended to anyone of any age." Computing Today
"Without question the best general introduction to computing I have ever seen." Personal Computer World
". . . perhaps the best introduction around . . . outstanding . . ." Educational Computing
"These books are outstanding . . . they make all other young people's computer books look meretricious." Times Educational Supplement

Guide to Computers
Understanding the Micro
Computer and Video Games
Computer Jargon
Computer Graphics
Inside the Chip
Computer Programming
Practise Your BASIC
Better BASIC
Machine Code for Beginners
Practical Things to do with a Microcomputer
Computer Spacegames
Computer Battlegames
Write Your Own Adventure Programs
Creepy Computer Games

New Titles

Programming Tricks & Skills Professional tips and tricks for better programming.

Experiments With Your Computer An exciting introduction to scientific investigation with a microcomputer.

Expanding Your Micro A detailed guide to add-ons and interfaces.

Write Your Own Fantasy Games A step-by-step guide to writing fantasy games, with program listing.

Weird Computer Games, Computer Spy Games Short listings to run on most main home computers.

Mystery of Silver Mountain, Island of Secrets Unique adventure game books containing full program listings.

ISBN 0-86020-835-4

9 780860 208358

£4·50